UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Uroš Podobnikar

UPRAVLJANJE KAKOVOSTI IN ČIŠČENJE PODATKOV

MAGISTRSKO DELO

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Uroš Podobnikar

UPRAVLJANJE KAKOVOSTI IN ČIŠČENJE PODATKOV

MAGISTRSKO DELO

MENTOR: izr. prof. dr. Marjan Krisper

Ljubljana, 2016

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Večna pot 113
1000 Ljubljana, Slovenija
telefon: 01 47 98 100
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Številka: 150-MAG-ISO/2016
Datum: 29. 02. 2016

Uroš PODOBNIKAR, univ. dipl. inž. rač. in inf.

L j u b l j a n a

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Upravljanje kakovosti in čiščenje podatkov**

Data quality management and data cleaning

Tematika naloge:

V nalogi raziščite področje čiščenja podatkov ter njegov širši okvir – umestitev v upravljanje kakovosti podatkov. Določite umestitev teh področij v širši okvir upravljanja IT. Predstavite ogrodja, standarde in morebitne druge podlage, ki lahko organizacijam nudijo vodilo in smernice za upravljanje (kakovosti) podatkov. Predstavite pojme – termine, ki so na tem področju pomembni ter njihovo medsebojno zvezo. Pojasnite tudi morebitno povezanost omenjenega področja s področjem interneta stvari (IoT – "Internet of Things").
Predstavite problematiko in raziščite razloge, zaradi katerih v organizacijah prihaja do napak v podatkih ter njihove posledice na organizacijo. Predstavite različne tipe napak.
Naredite pregled morebitnih aplikacij s tega področja, ki so na voljo na trgu.
V praktičnem delu predstavite predlog za izboljšanje stanja v organizacijah na primeru izdelave primera oz. prototipa programske rešitve za realizacijo tistega dela upravljanja s podatki, ki se nanaša na vzdrževanje pravilnosti in skladnosti podatkov (t.i. »data cleaning«).
Na podlagi raziskanega področja predstavite morebiten svoj predlog za izboljšanje stanja v organizacijah.

M e n t o r :

izr. prof. dr. Marjan Krisper

Fakulteta za računalništvo in informatiko
UNIVERZA V LJUBLJANI
2

D e k a n :

prof. dr. Nikolaj Zimic

Zahvala

Zahvaljujem se mentorju izr. prof. dr. Marjanu Krisperju za mentorstvo, strokovno pomoč, nasvete in usmeritve pri izdelavi magistrskega dela.

Hvala sodelavcu Alešu Miku za razlago strežniške infrastrukture ZPIZ.

Zahvaljujem se tudi sestri Mojci in tašči Emi za lektoriranje besedila in lektorske nasvete ter ge. Zdenki Velikonja s Fakultete za računalništvo in informatiko za pomoč in navodila pri administrativnih zadevah.

Posebna zahvala gre moji družini, še posebej ženi Mariji, za razumevanje in podporo v času izdelave magistrskega dela.

Eli in Filipu

Kazalo

Kazalo slik

Kazalo preglednic

Seznam uporabljenih kratic

kratica	angleško	slovensko (prevod ali pomen)
3NF	Third Normal Form	tretja normalna oblika
ACL	Access Control List	seznam za kontrolo dostopa
ANSI	American National Standards Institute	ameriški državni inštitut za standarde
BDD	Business Data Dictionary	podatkovni slovar organizacije
BPM	Business Process Management	upravljanje poslovnih procesov
CDQM	Corporate Data Quality Management	ogrodje za upravljanje kakovosti podatkov organizacije
CDS	Chief Data Steward	vodja skrbništva podatkov
CEO	Chief Executive Officer	generalni, izvršni direktor
CFD	Conditional Functional Dependencies	pogojne funkcijske odvisnosti
CIO	Chief Information Officer	direktor oddelka za informatiko
COBIT	Control OBjectives for Information and related Technology	ogrodje za obvladovanje IT
COIB	Cognitive Oriented IoT Big-data framework	ogrodje za obravnavo podatkov interneta stvari
CRLF	Carriage-Return Line-Feed	ukaz za vnos nove vrstice
CRP		Centralni Register Prebivalstva
CRUD	Create, Read, Update, Delete	vnos, branje, sprememba, brisanje
D^2Q	Data and Data Quality	model podatkov in kakovosti podatkov
DBA	DataBase Administrator	administrator podatkovne baze
DBMS	DataBase Management System	sistem za upravljanje podatkovnih baz
DDL	Data Definition Language	jezik za določitev podatkovnih struktur
DMBOK	The DAMA Guide to the Data Management Body of Knowledge	zbirka najboljših praks in priporočil na področju upravljanja podatkov
DML	Data Manipulation Language	jezik za delo s podatki
DQM	Data Quality Management	upravljanje kakovosti podatkov
EMRIS		Enotna Metodologija Razvoja Informacijskih Sistemov
ER	Entity Relationship	model povezav med entitetami
ETL	Extraction, Transformation, Loading	proces pridobitve, transformacije in nalaganja podatkov
IAM	Identity Access Management	upravljanje identitet za dostop

IEC	International Electrotechnical Commission	mednarodna elektrotehnična komisija za standardizacijo
IoT	Internet of Things	internet stvari
IP	Internet Protocol	internetni protokol
ISMS	Information Security Management System	sistem za upravljanje informacijske varnosti
ISO	International Organization for Standardization	mednarodna organizacija za standardizacijo
IT	Information Technology	informacijska tehnologija
ITIL	Information Technology Infrastructure Library	zbirka najboljših praks na področju storitev IT
LAN	Local Area Network	lokalno omrežje
PCI DSS	Payment Card Industry Data Security Standard	varnostni standard na področju kartičnega poslovanja
PL/SQL	Procedural Language extension to Structured Query Language	razširitev SQL s proceduralnim jezikom
PRS		Poslovni Register Slovenije
RAID	Redundant Array of Independent Disks	redundantno diskovno polje
RFID	Radio Frequency IDentification	radiofrekvenčno prepoznavanje
SCADA	Supervisory, Control And Data Acquisition	nadzor, krmiljenje in zajem podatkov
SLA	Service Level Agreement	sporazum – dogovor o ravni storitve
SMTP	Simple Mail Transfer Protocol	protokol za prenos elektronske pošte
SoD	Segregation of Duties	razdelitev dolžnosti
SQL	Structured Query Language	strukturiran povpraševalni jezik
SSL	Secure Sockets Layer	sloj varnih vtičnic
TLS	Transport Layer Security	varnost prenosnega sloja
UDF	User-Defined Function	uporabniško določena funkcija
UUP		Uredba o Upravnem Poslovanju
VSAM	Virtual Storage Access Method	metoda dostopa do datotečno organiziranih podatkov
WAS	Websphere Application Server	aplikacijski strežnik Websphere
XML	eXtensible Markup Lanuage	razširljivi označevalni jezik
z/VM	z Virtual Machine	operacijski sistem podjetja IBM
ZPIZ		Zavod za Pokojninsko in Invalidsko Zavarovanje
ZVOP		Zakon o Varstvu Osebnih Podatkov

Povzetek

Današnje organizacije se pogosto soočajo z izzivom, kako obvladovati veliko količino podatkov, ki jih uporabljajo pri svojem poslovanju. Zaradi mnogih razlogov je zelo pomemben vidik obvladovanja podatkov tudi zagotavljanje in ohranjanje ustrezne kakovosti podatkov. V organizacijah namreč po eni strani ustrezno visok nivo kakovosti podatkov predstavlja konkurenčno prednost, po drugi strani pa slaba kakovost podatkov vodi v številne neljube posledice.

V preteklosti so se izoblikovala ogrodja, metode ter orodja kot pomoč pri zagotavljanju ustrezne ravni kakovosti podatkov, poleg tega je kakovost podatkov obravnavana tudi v različnih standardih in zakonodaji. Kljub temu pa raziskave kažejo, da je stanje v organizacijah na tem področju še vedno razmeroma slabo.

Namen naloge je raziskati in predstaviti področje kakovosti podatkov v organizacijah ter predstaviti problematiko, ki iz tega izhaja. Predstavljene so posledice slabe kakovosti podatkov ter vzroki, ki vodijo v takšno stanje. Podani so tudi razlogi, zakaj je kakovost podatkov v organizacijah pomembna, ter predstavljeni standardi in zakonodaja s tega področja. Problematika kakovosti podatkov se pojavlja tudi na področju interneta stvari, ki je v zadnjem času deležno velikih raziskovalnih prizadevanj, zato je obravnavano področje prikazano tudi iz tega zornega kota.

V nalogi je največji poudarek na tistem delu področja, ki se nanaša na kakovost in čiščenje obstoječih podatkov. Predstavljene so vrste napak, različna ogrodja čiščenja podatkov ter prikaz postopka z združenimi poudarki različnih ogrodij. Narejen je tudi pregled obstoječih programskih rešitev s tega področja. Omenjeno je predstavljeno v prvem, teoretičnem delu naloge. Drugi del predstavlja praktični del, kjer je podan predlog za izboljšanje stanja v organizacijah s pomočjo izdelane programske rešitve – prototipa za realizacijo tistega dela upravljanja s kakovostjo podatkov, ki se nanaša na vzdrževanje pravilnosti podatkov s pomočjo zaznavanja napak v podatkih in možnost njihove odprave. Podan je tudi predlog uporabe rešitve v konkretni organizaciji s predlogom umestitve v obstoječi informacijski sistem z upoštevanjem vodil in principov, ki jih predlaga literatura.

V zaključnem delu naloge so podani ključni pristopi, ki bi v organizacijah pripomogli k izboljšanju stanja na tem področju.

Ključne besede: kakovost podatkov, celovitost podatkov, upravljanje kakovosti podatkov, upravljanje podatkov, čiščenje podatkov, informacijska varnost

Abstract

Today´s enterprises are often challenged by managing a large amount of data used in their business operation. Assurance and maintenance of adequate data quality level are important aspects of data quality management due to many reasons. On the one hand, the adequate data quality level represents a competitive advantage, and on the other hand, low data quality level leads to many unpleasant consequences.

In the past, frameworks, methodologies, and tools to help ensuring adequate level of data quality were formed. Besides, the question of data quality is discussed in legislation and various standards. Despite that fact, some researches show poor state of data quality in enterprises.

A purpose of the thesis is to research and present the area of data quality, and to show subsequent issues of low data quality. The thesis presents consequences as well as reasons of low data quality. It also shows reasons of data quality importance. In addition, it presents standards, legislation, and best practices that deal with the field of data quality. Data quality issues also arise in the field of the Internet of Things, which is an object of many researches lately, therefore, the thesis also presents main issues from that point of view.

The main emphasis of the thesis is on the part of the field dealing with data quality and data cleaning. The thesis presents error types, various data cleaning frameworks, and combines their main activities in a consolidated view. Furthermore, the thesis presents an overview of the existing software solutions available on the market to support data cleaning tasks. The aforementioned is introduced in the theoretical part of the thesis. The second part of the thesis represents a practical part, where a proposal for data quality improvement is given using a prototype of a software solution to address a specific part of data quality management, which deals with data accuracy maintenance by sensing errors in data, and the possibility of error elimination (data cleaning). In addition, the thesis proposes installation of the solution in a concrete organisation´s information system by considering principles and rules the literature suggests.

In the conclusion, there are essential approaches given to aid the improvement of data quality field in enterprises.

Key words: data quality, data integrity, data quality management, DQM, data management, data cleaning, information security

1. Uvod

1.1 Problematika kakovosti podatkov

Kakovost podatkov je pomemben del poslovnega sistema in vpliva na uspešno prilagajanje organizacije zunanjim, tržnim zahtevam [41]. Kljub temu se zelo pogosto dogaja, da se kakovosti podatkov v organizacijah ne posveča dovolj pozornosti, saj je raven zavedanja pomena kakovosti podatkov nizka, na kar kažejo številni avtorji in raziskave, kot je podrobneje prikazano v točki 4.

Slaba kakovost podatkov ima lahko negativne posledice na več ravneh. Z njimi se srečujemo zaposleni v IT, pogosto pa nepravilnosti v podatkih zazna šele končni, poslovni uporabnik aplikacij, kar lahko povzroča zastoje v poslovnih procesih, daljše trajanje poslovnih procesov, nezadovoljstvo uporabnikov in strank, napačno delovanje aplikacij ter ostale posledice, ki jih opisuje literatura [6, 18, 20] ter so podrobneje opisane v nadaljevanju. S slabo kakovostjo podatkov pa se posredno srečujejo tudi na vodstveni ravni. Podatki so v organizacijah namreč vir za odločanje, kar posledično pomeni, da slaba kakovost podatkov vodi tudi v napačne odločitve na višjih ravneh [19, 20].

Problemsko področje je v literaturi široko obravnavano, kakovost podatkov pa je prikazana z različnih zornih kotov in na različnih nivojih upravljanja, kot je predstavljeno v točki 3. Kakovost podatkov ni elementarna in enostavno merljiva lastnost. Definicija pojma kakovosti podatkov se je razvijala skozi obdobje raziskav. Kakovost je sestavljena iz različnih dimenzij [59], različni avtorji pa so jih nekoliko različno navajali. Vse to lahko kaže na težo problema, ki ga imajo organizacije pri zagotavljanju celostne obravnave kakovosti.

Organizacije imajo pri obravnavi problematike in zagotavljanju ustrezne ravni kakovosti podatkov na voljo več prijemov. V nekaterih državah je področje do določene mere celo urejeno z zakonodajo, kar še posebej velja za finančne podatke [20]. Na voljo so različni standardi, dobre prakse, njihova vpeljava pa je zahtevna [38]. Nadalje imajo na voljo različna ogrodja, izoblikovalo se je upravljanje kakovosti podatkov.

Običajni pogled na napake in nepravilnosti je ta, da so slabi. Vendar po drugi strani lahko razumevanje napak in njihovo širjenje vodi do aktivne kontrole kakovosti in izboljšanja upravljanja kakovosti podatkov [7].

1.2 Namen in cilji naloge

Namen naloge je raziskati področje in problematiko slabe kakovosti podatkov v organizacijah, še posebej tistega dela, ki se nanaša na kakovost in čiščenje obstoječih podatkov. Različni avtorji navajajo nekoliko različne poudarke v metodologiji čiščenja podatkov. Iz različnih pregledanih ogrodij je v nalogi izdelan enoten prikaz postopka čiščenja podatkov. V nalogi je podan tudi predlog za izboljšanje stanja v organizacijah s pomočjo izdelane programske rešitve oz. prototipa za realizacijo tistega dela upravljanja s kakovostjo podatkov, ki se nanaša na vzdrževanje pravilnosti, konkretneje zaznavanje napak v podatkih in možnost njihove odprave. Predstavljen je predlog uporabe rešitve v konkretni organizaciji in umestitve v obstoječi informacijski sistem.

Pri obravnavi področja moramo biti pozorni tudi na terminologijo, saj se termini med seboj dopolnjujejo in povezujejo v celoto oz. definicijo drugih terminov. Zato bo predstavljena tudi pomembnejša terminologija tega področja.

Motivacija za obravnavo navedene teme izhaja iz mojega dosedanjega dela v organizaciji, kjer sem zaposlen. Zaposleni v IT se pri delu s podatki v širšem pomenu (prenosi podatkov iz podedovanega sistema v relacijsko bazo, analize podatkov v relacijski bazi, delo s podatki v podatkovnem skladišču, izdelava enotnih registrov organizacije, izdelava aplikacij, ki te podatke uporabljajo, izmenjave podatkov z zunanjimi institucijami itd.) pogosto srečujemo s težavami glede kakovosti podatkov. Pri tem gre za različne vrste težav - neskladnost podatkov, strukturne težave v bazi, nekontrolirana rast posameznih tabel v smislu števila zapisov, napačne vrednosti v zapisih in podobno. S takšnimi težavami pa se ne srečujemo samo zaposleni v IT, ampak tudi poslovni uporabniki aplikacij.

1.3 Struktura naloge

Vsebina naloge je naslednja: prvi del je teoretičen in širše predstavi problemsko področje. Najprej so predstavljene posledice pomanjkljive kakovosti podatkov. Nato so opredeljeni pomembnejši termini tega področja ter povezave med njimi. Sledi razlaga, zakaj je kakovost podatkov v organizacijah pomembna, kakšni so pristopi k izboljšanju stanja ter na kakšen način se upravljanje kakovosti podatkov umešča v organizacije. V zadnjem času se veliko raziskovalnega dela vlaga v področje interneta stvari, kjer problematika prav tako zaseda pomembno mesto, zato naloga predstavlja tudi problematiko na tem področju. V naslednjem poglavju sledi predstavitev vzrokov nepravilnosti v podatkih, nato pa so predstavljeni standardi, zakoni in najboljše prakse na tem področju. Njihova vpeljava ni enostavna, zato so predstavljeni tudi izzivi, s katerimi se organizacije pri tem soočajo. Šesto poglavje predstavlja

reaktivni pristop k reševanju problematike, torej odpravo obstoječih napak ali čiščenje podatkov. Predstavljene so vrste napak, različna ogrodja čiščenja podatkov, metrike ter obstoječe programske rešitve za namen čiščenja podatkov.

Drugi del naloge je praktični. Predstavljena je programska rešitev oz. prototip, ki sem ga izdelal za namen zaznave napak v podatkih. Podan je predlog za razširitev programske rešitve z namenom uporabe tudi za čiščenje podatkov ter opisovanje podatkov ter umestitev v informacijski sistem organizacije, kjer sem zaposlen. Nazadnje je podan še zaključek.

2. Posledice pomanjkljive kakovosti podatkov

Kakovost podatkov ni elementarna lastnost, ampak je sestavljena iz več komponent ali dimenzij, kot je razloženo v točki opredelitve pojmov. Del literature navaja posledice nepravilnosti določenih komponent kakovosti podatkov, na primer celovitosti, del pa navaja posledice pomanjkljive kakovosti podatkov.

Boritz [6] navaja trditev več avtorjev [5, 47, 60], da je vpliv nepravilnosti v celovitosti podatkov in informacij za organizacije daljnosežen in terja veliko porabo sredstev, časa in ostalih virov, hkrati pa ima negativen učinek na ugled in odvrača stranke. Avtor kot primer navaja odmeven primer Fannie Mae z velikimi negativnimi finančnimi posledicami, katerih vzrok je bil ravno v omenjenih napakah. Primer s podobnimi posledicami, ki se je zgodil v banki Société Générale leta 2008, navaja tudi Gelbstein [20]. Finančne posledice navaja tudi Geiger [19] – po podatkih z Data Warehousing Institute naj bi slaba kakovost podatkov ameriška podjetja letno stala šeststo milijard ameriških dolarjev. Avtor še navaja, da je slaba kakovost podatkov tudi pogost vzrok neuspeha IT projektov.

Watts, Shankaranarayanan in Even [62] navajajo naslednje posledice slabe kakovosti podatkov:

- zmanjšane sistemske zmogljivosti,
- zmanjšano uporabnost sistema,
- napačne odločitve,
- zmanjšanje ugleda,
- večjo izpostavljenost tveganju,
- finančne izgube.

Skrajna primera z najhujšimi posledicami, ki sta se zgodila zaradi slabe kakovosti podatkov, sta eksplozija vesoljskega plovila Challenger in sestrelitev iranskega potniškega letala [18].

Posledice nepravilnosti v podatkih lahko na nivoju organizacije strnemo v naslednje točke:

- zastoji v poslovanju,
- daljše trajanje poslovnih procesov,
- zmanjšane sistemske zmogljivosti,
- nezadovoljstvo strank v postopkih,
- nezadovoljstvo zaposlenih,
- zmanjšanje ugleda,
- finančne posledice,
- neuspeh IT projektov.

3. Kakovost podatkov

3.1 Opredelitev pojmov

Termini, ki so uporabljeni v nalogi, so v pregledani literaturi opredeljeni na več mestih, pogosto pa si definicije med seboj niso popolnoma enotne (kar bo v nadaljevanju predstavljeno kot del problematike) ali pa je isti izraz uporabljen na različne načine (primer je skladnost). V nadaljevanju je podana razlaga terminov, ki se nanašajo na podatke in informacije in se pojavljajo v povezavi s kakovostjo podatkov. Prikazana je tudi povezava med nekaterimi.

Podedovani sistemi (ang. *legacy systems*)
Podedovani sistemi so zastareli računalniški sistemi, ki so zaradi svojega pomena še vedno v uporabi [79].

Umazani podatki (ang. *dirty data*)
Umazani podatki so podatki, za katere smo odkrili, da vsebujejo kakršne koli napake [45].

Redundanca podatkov (ang. *data redundancy*)
Redundanca je pojavljanje istega podatka na več mestih – na več fizičnih pogonih ali v več tabelah v podatkovni bazi [27]. Razloga sta največkrat varnost in zmogljivost. Lahko pa gre tudi za nepravilno načrtovanje podatkovnega modela, kar na primerih pokaže [32].

Sočasnost podatkov (ang. *data concurrency*)
Sočasnost pomeni, da lahko do istega podatka v istem trenutku dostopa več uporabnikov [89].

Skladnost podatkov (ang. *data consistency*)
Podatki so skladni ali konsistentni, kadar vsi uporabniki vidijo enak, usklajen podatek, kljub spremembam, ki so jih naredile transakcije uporabnikov [72]. Izraz se pogosto uporablja tudi za ujemanje vrednosti podatkov v primeru redundance [59]. Kot ugotavlja [67], avtorja Moerkotte in Lockemann [37] skladnost enačita s celovitostjo – podatkovna baza naj bi bila skladna, če je njeno stanje odraz spoštovanja nabora pravil in pogojev. Pri tem navaja delitev skladnosti na notranjo in zunanjo. Notranjo skladnost se lahko doseže z uporabo konceptov, ki jih v osnovi ponujajo podatkovne baze, in s pravilnim načrtovanjem podatkovnih modelov. Zunanja skladnost pa se doseže z uporabo definiranih pravil in pogojev v obliki omejitev (ang. *consistency constraints*).

Skladnost se uporablja tudi v kontekstu ujemanja s standardi in pravili [6].

Skladnost torej opisuje stanje v bazi podatkov in ne lastnosti v odnosu do objekta v realnem svetu, ki ga opisuje.

Natančnost ali pravilnost podatkov (ang. *data accuracy/correctness*)
Natančnost podatkov pomeni, da podatki opisujejo objekt v realnosti z ustrezno mero preciznosti oz. točnosti. Natančnost je v tesni zvezi s pravilnostjo (imenujemo jo tudi točnost) in se ju v nekaterih področjih enači [6].

Poznamo dve metodi za izračun pravilnosti [51]:

- pri prvi je rezultat katera koli vrednost med 0 in 1, pri čemer 1 pomeni točno ujemanje [12].

$$Pravilnost(v) = 1 - \left[\frac{razdalja(v, v')}{max(|v|, |v'|)}\right] \quad (1)$$

  pri tem je **v** izmerjena vrednost, **v'** pa dejanska pravilna vrednost.
  Funkcija *razdalja* pomeni število korakov (vstavkov, odstranitev, nadomeščanj znakov) [4]. Npr. če je *v = Jaez* in *v′ = Janez*, potem je *razdalja(v, v') = 1* .
  Imenovalec $max(|v|, |v'|)$ pa označuje največjo možno razdaljo med vrednostima.

- pri drugi sta možna rezultata dva: 1 in 0, pri čemer 1 pomeni točno ujemanje, 0 pa neujemanje [4].

$$Pravilnost(v) = \begin{cases} 1, & v = v' \\ 0, & v <> v' \end{cases} \quad (2)$$

  pri tem je **v** izmerjena vrednost, **v'** pa dejanska pravilna vrednost.

Popolnost podatkov (ang. *data completeness*)
Popolnost podatkov pomeni, da so vrednosti atributov vnesene [19, 59]. Omejitve pri meritvah in obdelovanju podatkov v sistemu onemogočajo stoodstotno popolnost v realnem času. Še posebej to velja za tiste objekte spremljanja, ki se pogosto spreminjajo. Posledično to tudi onemogoča stoodstotno pravilnost podatkov. V času trajanja transakcije podatek v podatkovni bazi ni popoln. Stopnja popolnosti, ki je dosežena, določa zgornjo mejo stopnje pravilnosti, ki je lahko dosežena [6].

Pravočasnost podatkov (ang. *data timeliness*)
Boritz [6] meni, da je absolutno popolnost in pravilnost podatkov težko doseči na racionalen način. To utemeljuje s trditvijo, da na pravočasnost podatkov vplivajo spremembe v realnem svetu, ki ga podatki opisujejo, in zastoji pri obdelavi podatkov s sorazmernim vplivom tudi na pravilnost podatkov. Ker je čas kontinuiran, moramo popolnost in pravilnost razumeti v okviru sprejemljivih mej, ki določajo pravočasnost podatkov in posledično pravilnost [6]. Avtor ob tej definiciji dodaja še, da je lahko pravočasnost podatkov okrnjena tudi zaradi obdelave podatkov. Različni deležniki imajo lahko pri tem različne tolerance, kdaj je podatek še pravočasen. Zaradi tega je uporaben koncept časovnega žigosanja. Ko je podatek opremljen s časovnim žigom, je pravilnost lažje preverljiva.

Enačba za izračun pravočasnosti [1] :

$$Pravočasnost(v,s) = \max\left[\left(1 - \frac{Trenutnost(v)}{Nestabilnost(v)}\right); 0\right]^{s} \quad (3)$$

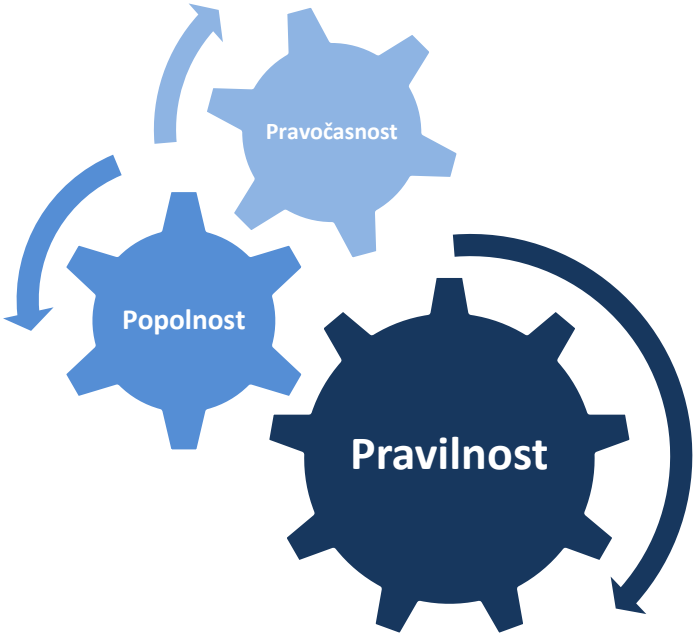
Parameter *s* služi za nastavitev občutljivosti pravočasnosti na razmerje med trenutnostjo in nestabilnostjo. Vrednost se izbere v odvisnosti od okoliščin. Pri manjši vrednosti (npr. 0,5) ima parameter na pravočasnost manjši vpliv, pri večji vrednosti (npr. 2) je vpliv večji, pri vrednosti 1 vpliva ni.

Nestabilnost (ang. *volatility*) je definirana kot dolžina časa, v katerem podatek ostane veljaven, *trenutnost* (ang. *currency*) pa je starost podatka, ko je dostavljen uporabniku, izračun pa je naslednji [51]:

$$Trenutnost(v) = \text{ČasDostave}(v) - \text{ČasZadnjePosodobitve}(v) + \text{Starost(v)} \quad (4)$$

$$Nestabilnost(v) = \text{ČasIzteka}(v) - \text{ČasZadnjePosodobitve}(v) + \text{Starost(v)} \quad (5)$$

Boritz [6] pravočasnost, popolnost in pravilnost povezuje na naslednji način: pravočasnost pogojuje popolnost, popolnost pa določa zgornjo mejo pravilnosti podatkov (slika 1).



Slika 1: Povezava treh lastnosti

Veljavnost podatkov (ang. *data validity*)
Koncept veljavnosti se uporablja za podatke, ki opisujejo neotipljive objekte, in pomeni, da podatek pravilno predstavlja pogoj, pravilo ali razmerje. Veljavnosti se torej ne uporablja za podatke, ki opisujejo fizične objekte. V splošnem so pogoji, pravila ali razmerja veljavni, če je resnično, kar opisujejo. V poslovnem kontekstu so transakcije veljavne, če so bile sprožene in izvedene s strani osebja ali sistema z ustreznimi pooblastili in če je dovoljenje pristno in znotraj obsega pooblastil izdajatelja dovoljenja [6].

Ugled podatkov (ang. *data reputation*)
Ugled podatkov je odvisen od vira. Običajno imajo viri podatkov z dolgo tradicijo boljši ugled [51].

$$OcenaAtributa\ (s,a) = \frac{\sum_{j=1}^{m} Ocena[a,j]}{m} \tag{6}$$

$$Ugled(s) = \sum_{a=1}^{n} Teža[a] * OcenaAtributa(s,a) \tag{7}$$

Pri tem *OcenaAtributa(s,a)* označuje skupno oceno za ugled atributa *a*, izračunanega kot povprečje vseh razpoložljivih ocen *Ocena*[*a,j*] za ta atribut *a*. Spremenljivka *m* pomeni število uporabnikov, ki ocenjuje vir, *s* pa je vir, ki se ocenjuje. *Teža*[*a*] pomeni utež posameznega

atributa in temelji na pomembnosti glede na ostale atribute, *n* pa pomeni število atributov [51].

Varnost podatkov (ang. *data security*)
Podatkovna varnost s fizičnimi in programskimi kontrolami dostopov onemogoča nepooblaščen dostop do podatkov z namenom varovanja podatkov pred naravnimi nesrečami ter pred namernimi in nenamernimi zlorabami podatkov (nepooblaščeni vnos, spremembe, uničenje), ki bi lahko ogrozile celovitost podatkov [6].

Razpoložljivost ali dostopnost podatkov (ang. *data availabilty/accessibility*)
Lastnost informacijskega sistema, da v določenem trenutku zagotavlja dostop do podatkov [79]. Podatki morajo biti uporabnikom na voljo v obliki, ki je zanje uporabna, v skladu s poslovnimi zahtevami. Nerazpoložljivi podatki imajo negativne posledice na kakovost uporabnikovih aktivnosti in odločitev [40]. Razpoložljivost podatkov je komplementarna podatkovni varnosti. Podatkovna varnost onemogoča nepooblaščen dostop do podatkov, razpoložljivost podatkov pa olajša pooblaščeni dostop do podatkov [6].

Preverljivost podatkov (ang. *data verifiability/auditability*)
Podatki so preverljivi takrat, kadar lahko neodvisni opazovalec z uporabo enakega postopka in enake tolerance popolnosti, pravilnosti, pravočasnosti in veljavnosti dobi enak rezultat. To lastnost angleško imenujemo *verifiability*. *Auditability* pa se nanaša na možnost sledljivosti podatka do njegovega vira, s čimer ga potrdimo ali ovržemo. Preverljivost podatkov predstavlja potreben pogoj za zagotovitev celovitosti [6].

Verodostojnost podatkov (ang. *data credibility/assurance*)
Fizična neoprijemljivost podatkov uporabnikom omejuje možnost ocenjevanja celovitosti [48, 58]. Za zaupanje v celovitost podatkov mora obstajati dokaz, da je bila zaščitena pred ponarejanjem in poseganjem nepooblaščenih oseb [6]. Avtor še dodaja, da preverljivost podatkov predstavlja potreben pogoj za zagotovitev celovitosti, medtem ko verodostojnost izvira iz dejansko uporabljenih postopkov za preverjanje celovitosti.

Tajnost podatkov (ang. *data secrecy*)
Tajnost podatkov je lastnost podatkov, ki jo dosežemo s preprečevanjem nepooblaščenega dostopa ali razkritja [67].

3.1.1 Celovitost podatkov (ang. data integrity)

Celovitost je izmed vseh predhodno obravnavanih terminov v pregledani literaturi najbolj razdelana in največkrat obravnavana, zato je pojasnjena v posebni točki. Celovitost imenujemo tudi neokrnjenost. Literatura navaja vrsto definicij, hkrati pa je celovitost v literaturi obravnavana v različnih kontekstih:

- na nivoju zapisovanja in branja z medija,
- v kontekstu podatkovne baze,
- na nivoju informacij, podatkov,
- glede na zorni kot opazovanja – vlogo.

Nivo zapisovanja in branja z medija

Na nivoju shranjevanja podatkov oz. zapisovanja in branja z medija definicijo podaja [21], ki navaja, da je celovitost podatkov glavna skrb shranjevanja podatkov, celovitost pa pomeni, da so prebrani podatki enaki podatkom ob shranjevanju ali prenosu. Avtor predlaga novo tehniko za boljše odkrivanje napak v celovitosti podatkov na osnovi kontrolnega določitvenega dejavnika (ang. *Check Determinant Factor – CDF*), ki naj bi bila učinkovitejša od tradicionalnih metod, Hammingovega šifriranja in RAID.

Nivo podatkovne baze

Na nivoju podatkovne baze kot celote je celovitost po [27] stanje podatkovne baze, v katerem so omejitve (ang. *integrity constraints*) in pravila veljavna. Avtorja v [67] podata za bazni nivo podobno definicijo kot [21] – celovitost sistema zagotavlja, da so podatki, vneseni v sistem, po vsebini enaki tistim ob branju.

Avtorji [44] so na tem nivoju predlagali model zagotavljanja večnivojske tajnosti in celovitosti, ki naj bi bil bolj razumljiv in enostaven kot klasični večnivojski varnostni DBMS.

Nivo podatkov, informacij

Na nivoju podatkov je celovitost po [43] eden izmed ciljev informacijske varnosti, ki zagotavlja, da:

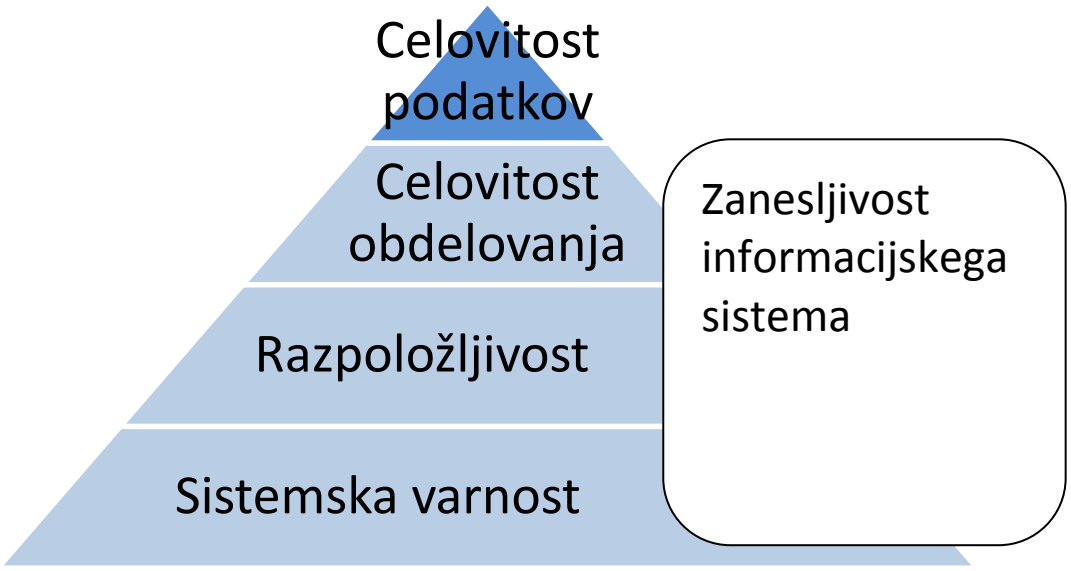
- je podatek pravilna predstavitev oz. preslikava informacije,
- podatek ohranja izvirni nivo natančnosti – pravilnosti,
- podatek ostaja nepoškodovan,
- podatek ob vnosu ni bil izpostavljen namenski ali nenamenski spremembi ali uničenju.

Celovitost se po [67] nanaša na preprečevanje nepooblaščenih sprememb podatkov.

Boritz [6] je izdelal pregled različnih definicij, ki so opisane v nadaljevanju. Navaja definicijo po COBIT-u (s katero se ne strinja v celoti in zanjo predlaga dopolnitev), kjer celovitost določajo tri lastnosti: **popolnost, pravilnost in veljavnost**. Druge definicije so tem atributom dodale še **pooblaščenost, pravočasnost, skladnost** in **ločitev nezdružljivih funkcij**. Primerjava različnih definicij v [6] je pokazala, da so lastnosti celovitosti v povezavi z **zanesljivostjo** (ang. *reliability*), **ustreznostjo** (ang. *relevance*), **uporabnostjo** (ang. *usability*), **kakovostjo** (ang. *quality*) in **vrednostjo** (ang. *value*). Celovitost je rezultat vseh navedenih lastnosti. V primerjavi s kakovostjo ima ožji pomen, pri čemer se nahaja v preseku treh glavnih lastnosti kakovosti (zanesljivosti, ustreznosti in uporabnosti), kot prikazuje slika 3 [6].

Avtor [6] navaja tudi usklajeno definicijo: **celovitost je zaupanje** (ang. *representational faithfulness*), **da podatek ali informacija podaja resnično stanje objekta, ki ga podatek ali informacija opisuje.** Pri tem je zaupanje sestavljeno iz štirih ključnih lastnosti: **popolnost, pravočasnost, pravilnost** in **veljavnost.** Te lastnosti dopolnjuje sedem dodatnih, sekundarnih lastnosti: **varnost, razpoložljivost, razumljivost** (ang. *understandability*), **primerljivost** (ang. *comparability*), **predvidljivost** (ang. *dependability, predictability*), **preverljivost** in **verodostojnost**. Omenjene štiri ključne lastnosti so nujne za zaupanje in posledično celovitost, medtem ko so sekundarne lastnosti v pomoč v določenih domenah, kadar je zaupanje izraženo z določeno mero in ne kot absolutno dosežena lastnost. Takrat lahko sekundarne lastnosti zvišajo mero zaupanja.

Slika 2 prikazuje, da celovitost podatkov temelji na zanesljivosti informacijskega sistema, ki jo sestavlja sistemska varnost, razpoložljivost in celovitost obdelovanja [6].



Slika 2: Temelji celovitosti podatkov

Glede na zorni kot opazovanja

Glede na zorni kot, s katerega opazujemo, avtor v [20] navaja več možnih definicij:

- za *varnostnega inženirja* celovitost podatkov pomeni, da podatki ne morejo biti spremenjeni neopaženo in da so spremenjeni le s strani tistih oseb ali sistemov, ki imajo ustrezno pooblastilo;
- za *administratorja* podatkovne baze celovitost pomeni, da so podatki, vneseni v podatkovno bazo, natančni ali pravilni, veljavni in skladni;
- za *podatkovnega arhitekta* celovitost pomeni, da so primarne entitete edinstvene, unikatne in določene (*not null*). To pomeni, da ni podvojenih entitet in da obstaja ključ, s katerim lahko dostopamo do vsake entitete;
- za *lastnika podatkov* je celovitost merilo kvalitete;
- za *prodajalca* je celovitost pravilnost in skladnost shranjenih podatkov, ki se kaže z odsotnostjo sprememb podatka med dvema običajnima posodobitvama podatkov, kar se doseže s samo zasnovo podatkovne baze.

Definicije celovitosti sta na več tipov razvrstila tudi Zviran in Glezer [67], vendar pa sta uporabila druge skupine:

- enoelementne podatkovno usmerjene definicije,
- enoelementne nepodatkovno usmerjene definicije,
- večelementne širše usmerjene definicije.

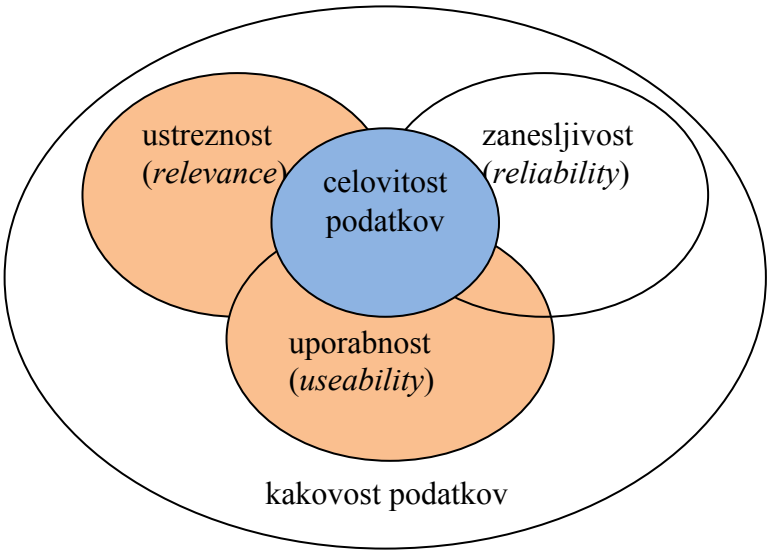
Avtorja v članku podajata definicije različnih skupin in različnih avtorjev. Izpostavita pa definicijo, ki naj bi celovitost najbolj obširno in popolno opisovala: **celovitost je lastnost, da podatki, informacijski proces, računalniška oprema, programska oprema, ljudje in ostale entitete ustrezajo pričakovani stopnji kvalitete, ki je zadovoljiva in zadostna v določenih okoliščinah.** Lastnosti kvalitete so lahko splošne ter odvisne od konteksta ali pa specifične, v skladu z načrtovano uporabo.

3.1.2 Kakovost podatkov (ang. data quality)

Preprosta definicija je naslednja: **kakovost podatkov je značilnost podatkov glede na pričakovane lastnosti** [79]. Ta definicija je nekoliko ohlapna, vendar pa Geiger [19] in Chapman [9] kakovost podatkov opisujeta na podoben način: **kakovost podatkov je zgolj primernost podatkov za uporabo in je relativni pojem**. Chapman jo označuje z besedno zvezo *"fitness for use"*, ki se v literaturi večkrat pojavi, npr. v [31, 41, 51, 62].

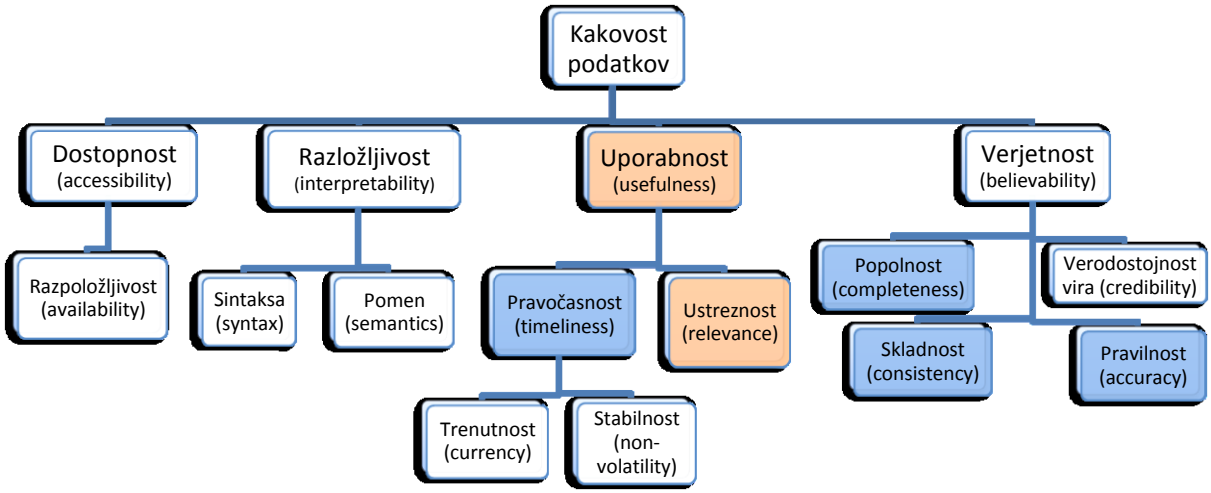
Boritz [6] pojasnjuje, da so tri glavne lastnosti kakovosti: **zanesljivost** (ang. *reliability*), **ustreznost** (ang. *relevance*) in **uporabnost** (ang. *usability*). Nujen gradnik kakovosti pa je

tudi **celovitost**. To ponazori z besedno zvezo *"... information integrity is the sine qua non of information quality ..."*, kar prikazuje slika 3 [6].



Slika 3: Povezava med celovitostjo in kakovostjo podatkov

Wang, Reddy in Kon [59] kakovost podatkov opisujejo kot večdimenzijski in hierarhični koncept, prikazan na sliki 4.



Slika 4: Dimenzije kakovosti podatkov

Na slikah 3 in 4 je z barvami označeno ujemanje med različnima opisoma kakovosti podatkov.

V literaturi pa imajo nekateri avtorji tudi drugačen pogled na odnos med celovitostjo in kakovostjo. Avtorja v [67] navajata, da Fernandez in ostali [16] celovitosti podatkov ne vidijo kot sestavni del kakovosti, ampak kot njej komplementarno lastnost. Navajata tudi ugotovitve Ruthberga in Polka [49], ki trdita, da celovitost ni enaka kakovosti, ampak je nabor posameznih lastnosti, pri katerem je množica teh lastnosti v celoti razumljena kot zadostna za določen namen.

Otto in ostali [41] so sklenili, da je skupni imenovalec definicij kakovosti ta, da je kakovost sestavljena iz več elementov, imenovanih tudi dimenzije kakovosti (kot so naštete v predhodnih opisih). Navajajo ugotovitev Wanga in Stronga [61], da je za ocenjevanje kakovosti treba oceniti vse različne dimenzije kakovosti.

V praksi je težko doseči popolno kakovost na celotni množici podatkov. Če želimo sto-odstotno pravilne in stoodstotno popolne podatke, je to lahko zelo drago ali celo ni vedno dosegljivo [6, 19]. Zato moramo pogosto sprejeti kompromis – v primeru napake moramo vedeti, ali nam je bolj pomembna popolnost ali pravilnost. V primeru, da je bolj pomembna pravilnost, bomo takšen zapis pri uporabi izpustili, v nasprotnem primeru ga uporabimo [19].

3.2 Pomen kakovosti podatkov

Zviran in Glezer [67] sta v letu 1999 zapisala svoj pogled na problematiko celovitosti podatkov. Po njunem mnenju podatki v svoji enostavni obliki posameznih zapisov (ang. *raw data*) izgledajo nepomembni, ko pa jih obravnavamo kot celoto, lahko tvorijo eno najbolj kritičnih prednosti organizacije in bi zato morali biti ustrezno upravljani in varovani. Dodajata, da so bile v preteklosti razvite osnovne tehnike varovanja podatkov, vendar je bil poudarek večinoma na tajnosti in razpoložljivosti podatkov, ne pa tudi na celovitosti. Sčasoma so podatkovni viri organizacij pridobivali na obsegu, kompleksnosti in vrednosti, zato se je pojavila potreba po mehanizmih za preprečevanje nepooblaščenega poseganja v podatke in po standardu podatkovne celovitosti, ki naj bi zagotavljal splošno merilo in orodja za vrednotenje različnih modelov in mehanizmov na tem področju. V ta namen predlagata Clark-Wilsonov model kot model podatkovne celovitosti in pa splošno uporabo definicije celovitosti, ki je zapisana v točki definicij celovitosti podatkov 3.1.1. Navajata namreč, da v tistem času ni bilo soglasja o uporabi enotne definicije, ki bi služila kot standard. Osnovni vzrok temu iščeta v pomanjkanju raziskovalne dejavnosti na tem področju. Po njunem obstajajo situacije, ko nepooblaščeno poseganje v podatke naredi več škode kot razkritje podatkov, zato takšno stanje celovitosti podatkov vzbuja veliko skrb pri uporabi informacijskih sistemov.

Splošno ogrodje za namen preverjanja celovitosti je v svoji raziskavi v letu 2005 predlagal in oblikoval tudi Boritz [6]. Navaja namreč, da so do takrat sicer obstajala ogrodja za kontrolo podatkov, vendar le v finančnih domenah.

Problematika celovitosti podatkov se sicer v letih po predhodno omenjenih raziskavah ni bistveno izboljšala. Pogled Zvirana in Glezerja na problematiko celovitosti v letu 2011 potrjuje Gelbstein [20], ki navaja, da so podatkovne baze najmanj zaščiteni objekti v vsej IT infrastrukturi. Kot možen vzrok temu navaja veliko različnih interpretacij in definicij podatkovne integritete, ki se med seboj prekrivajo, naslavljajo različne probleme in tako ustvarjajo zmedo v pomenu.

Omenjene poglede na problematiko potrjuje množica različnih definicij celovitosti v točki 3.1.1. Celovitost podatkov je po zapisanem v opredelitvi pojmov predpogoj za kakovost podatkov. Kakovost podatkov pa je ključni cilj učinkovitega, uspešnega obvladovanja organizacije [6]. Boritz navaja še trditev Weilla in Rossa [64], ki menita, da je kakovost podatkov najmanj razumljena in najslabše izkoriščena izmed vseh ključnih potencialnih organizacijskih prednosti.

Gelbstein [20] trdi, da dokler obvladovanju podatkov ne bo posvečena enaka mera pozornosti kot obvladovanju IT, toliko časa bodo organizacije izpostavljene precejšnjemu operativnemu in finančnemu tveganju, tveganju neskladnosti s predpisi in tveganju okrnjenega ugleda. Podobnega mnenja so P. Nastase, F. Nastase in Ionescu [38], ki poudarjajo, da je učinkovita uporaba IT za uspeh strategije celotne organizacije izrednega pomena, ker ima potencial za glavno gonilo ekonomskega uspeha v 21. stoletju. Podobno menijo Otto in ostali [41], ki navajajo da je učinkovito upravljanje podatkov glavni predpogoj za uspešno prilagajanje poslovnega modela spremenljivim tržnim zahtevam. Woodall, Borek in Parlikad [66] so celo mnenja, da je kakovost podatkov najpomembnejša za uspeh organizacije. Vendar pa kakovost podatkov težko obravnavamo kot absolutno in dokončno. Chapman [9] namreč trdi, da se napake v podatkih pojavijo kljub preprečevanju napak na vnosu. To pomeni, da tudi če zagotovimo celovitost podatkov na začetku njihovega življenjskega cikla, lahko do napak pride pozneje (slika 13). Suer in Nolan [55] menita, da se mora vodstvo organizacij zavedati, da ima skoraj vsak sistem določeno stopnjo slabih podatkov, pomembno pa je razumevanje vpliva teh podatkov na poslovanje in vzdrževanje takšnega nivoja pravilnosti podatkov, kot je sprejemljiv za poslovne uporabnike. Redman [46] trdi, da lahko pričakujemo 1 do 5-odstotno stopnjo napačnih podatkov, če le ni bilo izboljšanju namenjeno veliko truda.

3.3 Pristopi k reševanju problematike

Literatura obravnava področje kakovosti podatkov in njenih komponent, predvsem celovitosti, na več nivojih. Literatura glede na življenjski cikel podatkov predlaga metode, okvire in tehnike za obravnavo podatkov v več fazah življenjskega cikla (slika 13). Glede na literaturo in posamezne v nalogi opisane metode, okvire in tehnike, lahko strnemo pristope, ki pripomorejo h kakovosti podatkov, v naslednje skupine:

- skrb za kakovost v fazi načrtovanja IS, npr. [32, 59],
- skrb za kakovost ob zajemu, npr. [9, 32, 53, 56],
- skrb za kakovost po zajemu podatkov, npr. [21, 50].

Zadnjo skupino lahko nadalje delimo na:

- skrb za kakovost preko obstoječih procesov IT, npr. [20],
- skrb za kakovost preko organiziranega upravljanja kakovosti podatkov, npr. [15, 19, 41, 53, 55],
- organizirano čiščenje podatkov, npr. [9, 13, 26, 31, 45],
- *ad hoc* ročno čiščenje podatkov.

Posamezni primeri metod, ogrodij in tehnik so navedeni spodaj, nekateri pa so bili tudi že omenjeni v opredelitvi pojma celovitosti.

Ling, Goh in Lee [32] so oblikovali teoretično ogrodje za načrtovanje podatkovnega modela, ki je efektiven in praktičen, hkrati pa ne ogroža celovitosti podatkov v podatkovni bazi. To pomeni, da se ta pristop uporabi že med samim načrtovanjem podatkovnega modela. Glede na metodologijo EMRIS [25] je to v fazi načrtovanja informacijskega sistema ali dopolnitve obstoječega podatkovnega modela. Tudi Storey, Dewan in Freimer [56] menijo, da je potrebno probleme v kakovosti podatkov obravnavati čim bolj zgodaj – že v fazi načrtovanja. Ustrezen podatkovni model skupaj z omejitvami ima učinek ob zajemu podatkov v podatkovno bazo. Omenjeno ogrodje [32] temelji na treh novih normalnih oblikah: sproščeni, ponovljeni in sproščeni – ponovljeni tretji normalni obliki (ang. *relaxed 3NF, replicated 3NF, relaxed – replicated 3NF*), na osnovi močnih in šibkih funkcijskih odvisnosti ter dodatnih prijemov v izogib neskladnosti v podatkih; uporabo prevajalnikov z uveljavljanjem omejitev (ang. *constraint enforcement precompiler*) in prožilcev.

Rešitev na nivoju podatkovnega modela so predstavili tudi Wang, Reddy in Kon [59], ki predlagajo razširitev podatkovnega modela s kazalniki kakovosti na nivoju atributov posameznih zapisov, algebro za uporabo te razširitve ter pravili za enotno obravnavo kazalnikov kakovosti atributa in osnovnega atributa za namen zagotavljanja celovitosti.

Če zgornji pristop označimo kot klasičen, kjer mehanizem podatkovne baze zavrne neskladne podatke pred vnosom oz. ob poskusu vnosa v podatkovno bazo, pa je sledeči drugačen. Sadri [50] je predstavil pristop za upravljanje z negotovimi podatki – t. i. metodo sledenja podatkovnemu viru (ang. *Information Source Tracking – IST*), ki neskladnih podatkov ne zavrača, pač pa jih obravnava na poseben način. To pomeni, da so napake v podatkih obravnavane po vnosu. Avtor predstavi zgolj tehnični vidik na baznem nivoju, ni pa predloga uporabe takšnega pristopa v poslovnih procesih.

Chapman [10], Falge, Otto in Österle [15] ter Rahm in Do [45] izpostavljajo, da je za organizacije mnogo ugodneje, da napake v podatkih zaznajo čim prej, torej pred zapisom v podatkovno zbirko. Vendar pa je to možno le v nekaterih primerih (npr. interaktivni vnos posameznih podatkov), medtem ko je v primeru selitev podatkov iz podedovanega sistema to težje in je treba uporabiti čiščenje podatkov. Moramo pa se zavedati, da se napake v podatkih pojavijo kljub preprečevanju napak na vnosu, zato ne smemo pozabiti na poznejše potrjevanje in čiščenje podatkov [9]. Pomembno pri čiščenju podatkov je to, da sledimo določenim smernicam in ne uporabljamo *ad hoc* pristopa ročnega čiščenja podatkov, ker je težavno in časovno potratno, poleg tega je takšno čiščenje dovzetno za nove napake [34]. Menim pa, da se takšnemu načinu v praksi ne moremo povsem izogniti, kar potrjuje Chapman [9]. Pogosto prihaja do situacij, kjer uporabniki zaznajo določene napake v podatkih pri uporabi aplikacij in podatkov v produkcijskem okolju. Takšne napake je potrebno v sodelovanju uporabnikov in IT odpraviti nemudoma, saj ima njihova prisotnost negativne posledice na poslovni proces, uporabnik pogosto ne more nadaljevati s procesom.

Gelbstein [20] za izboljšanje stanja na področju celovitosti podatkov navaja t. i. pravilo:

- dveh D (*Detect, Deter*),
- dveh P (*Prevent, Prepare*),
- dveh R (*Respond, Recover*),

ali "odkrij, odvrni, preprečuj, pripravi, reagiraj, povrni". Koraki za zagotavljanje celovitosti podatkov naj bi naslavljali ta pravila. Pobuda mora biti s strani poslovnih uporabnikov, vloga IT pa je v izvedbi in vpeljavi postopka. Kot dobre prakse pa navaja naslednje aktivnosti:

- določitev lastništva podatkov in odgovornosti za celovitost (poslovni uporabniki),
- določitev pravic dostopov in pooblastil – upošteva se vodili potrebe po vedenju (ang. *need to know*) in najmanjšega nabora pooblastil (ang. *least privilege*),
- ločitev dolžnosti (ang. *Segregation of Duties – SoD*).

Unsworth in ostali [56] so 2011 kakovost podatkov prikazali v luči motivacije zaposlenih in njihove hierarhije ciljev. Menijo, da lahko s pravilnim razumevanjem psihološke plati dela s podatki, ciljev zaposlenih in njihovih povezav, organizacija sprejema ustrezne ukrepe, ki so

učinkovitejši kot določene kontrole. S podobnega zornega kota na kakovost podatkov gledajo Storey, Dewan in Freimer [53], ki menijo, da so zaposleni največje premoženje in prednost organizacije in da bi rešitve problematike morali iskati tako z organizacijske perspektive kot tudi s perspektive zaposlenih. Izpostavijo tri dejavnike za izboljšanje kakovosti:

- obravnava kakovosti že v fazi načrtovanja. Z uporabo referenčnih omejitev in omejitev celovitosti se veliko napakam izognemo že ob vnosu v sistem;
- ustrezno lastništvo podatkov. Lastnik podatkov bi moral postaviti politiko za uporabo in spremembe podatkov. Lastnik podatkov bi moral biti tista organizacijska enota, ki ima od podatkov največ koristi;
- sistem nagrajevanja. Za vzpostavitev takšnega sistema so potrebni predpogoji v obliki definicije kakovosti podatkov, katere značilnosti jih torej določajo, ter postavitev ciljev in metrik. Zaposleni, ki te cilje dosegajo, morajo biti ustrezno nagrajeni.

Uporabo prijemov za motivacijo zaposlenih za namen izboljšanja kakovosti podatkov predlagajo tudi Lee in ostali [28], pomembno mesto pa ima tudi v ogrodju za upravljanje kakovosti podatkov organizacije (v nadaljevanju CDQM), katerega avtorji so Otto in ostali [41].

Omenjeni avtorji so predlagali ogrodje CDQM, pozneje pa so Falge, Otto in Österle [15] predlagali metodo za izdelavo strategije takšnega upravljanja. V začetku vzpostavljanja upravljanja kakovosti podatkov (v nadaljevanju DQM) v organizacijah je bil pogosto poudarek predvsem na avtomatizaciji rešitev, ki so se nanašale na podatke strank [19].

Suer in Nolan [55] menita, da za upravljanje kakovosti podatkov potrebujemo:

- ljudi,
- postopke,
- orodja.

Opisujeta tudi njihovo povezanost; vodja skrbništva podatkov (ang. *Chief Data Steward – CDS*) potrebuje sistem za aktivno upravljanje kakovosti podatkov na več nivojih. Najprej morajo biti postavljena pravila za upravljanje podatkov od nastanka dalje, torej od prve faze življenjskega cikla podatkov. Ta pravila morajo postaviti poslovni uporabniki, njihov namen pa je zagotavljanje pravilnosti podatkov. Poslovni uporabniki potrebujejo tudi orodja, s katerimi spremljajo trenutno stanje kakovosti podatkov. Obstaja več tehnik za ocenjevanje kakovosti podatkov, ena izmed njih je [66], kjer avtor uporabi dinamičen pristop k ocenjevanju kakovosti z upoštevanjem najboljših praks na tem področju. Nekoliko drugačen pristop so predstavili Watts, Shankaranarayanan in Even [62], ki upoštevajo kognitivno perspektivo ocenjevanja kakovosti.

Ko je katero od pravil kršeno, mora poslovni uporabnik ukrepati. Avtorja [55] ob tem poudarjata, da to ni odgovornost IT vodstva, pač pa poslovnih uporabnikov. Omenjeni sistem mora biti sposoben v ozadju samodejno zaznati in odpraviti nekatere težave v podatkih, kot so: napačni naslovi, manjkajoči podatki in napačni podatkovni formati. Poiskati mora tudi redundanco v podatkih (v smislu podvojenih zapisov).

K zgornjim sredstvom bi po mojem mnenju morali dodati še metrike in pravila. Pomembnost metrik namreč izpostavlja Gelbstein [20] in je pojasnjena v točki 6.3. Pravila pa zaradi tega, ker se v praksi večkrat izkaže, da je bolje, če pravila niso del postopkov ali orodij (kot se razume iz zgornjega opisa postopka), pač pa so ločena in jih je mogoče uporabljati v več postopkih in jih hkrati aktivno upravljati brez sprememb postopkov ali orodij. V organizacijah se v ta namen uporabljajo sistemi poslovnih pravil.

3.4 Upravljanje kakovosti podatkov

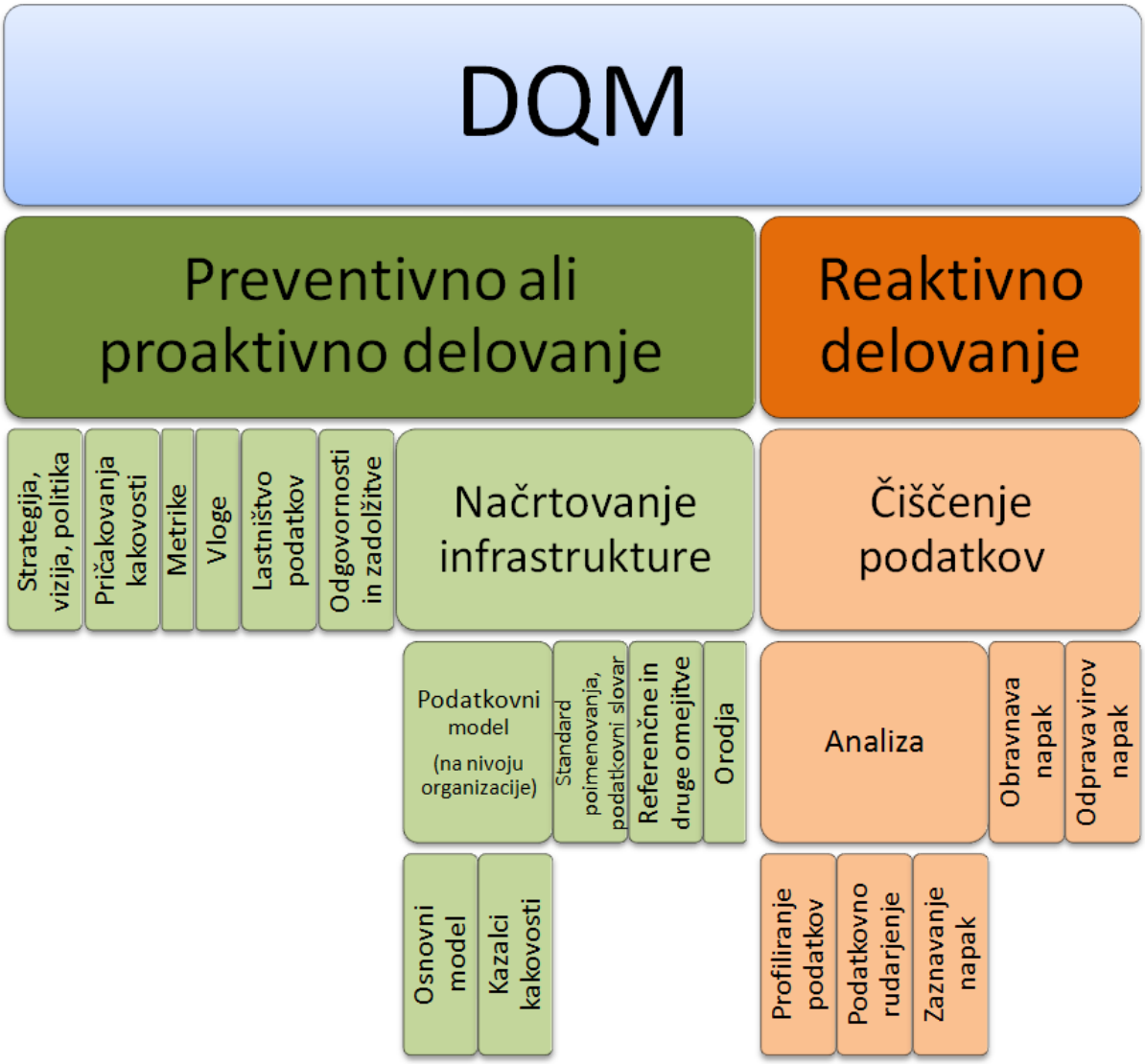
3.4.1 Umestitev DQM v organizaciji

Upravljanje kakovosti podatkov (ang. *Data Quality Management – DQM*) je v kakovost usmerjeno upravljanje podatkov in zajema zbiranje, shranjevanje, obdelavo, predstavitev, načrtovanje, razširjanje, organizacijo, uporabo in uničenje podatkov za podporo procesov poslovnega odločanja, operativnih procesov ter načrtovanje ustreznega okvira za neprekinjeno izboljševanje kakovosti podatkov [41, 63]. DQM zajema tudi vzpostavitev in uvedbo vlog, odgovornosti, politik in postopkov, ki se nanašajo na pridobitev, vzdrževanje, širitev in uničenje podatkov [19]. DQM združuje tako vidike upravljanja kakovosti kot tudi upravljanja podatkov, oboje pa se tipično umešča v upravljanje IT [41]. V literaturi je DQM obravnavana tudi kot organizacijska funkcija [15].

Spodnja slika 5 prikazuje odvisnost poslovnega odločanja od uspešnega upravljanja kakovosti podatkov [19]. Postavitev potrjuje princip, ki ga je na področju poslovne inteligence izpostavil Gelbstein [20] – *"garbage in, garbage out"*. To pomeni, da slabo upravljanje kakovosti podatkov vodi v slabo kakovost podatkov, ta pa v slabe poslovne odločitve. IT mora zagotoviti, da oseba, ki sprejema poslovne odločitve, pozna pomanjkljivosti v kakovosti podatkov ter kakšne so možnosti za njihovo odpravo. V nekaterih primerih so za odpravo potrebne spremembe poslovnih procesov.

Slika 5: Odvisnost poslovnega odločanja od DQM

Na podlagi pregledane literature na temo DQM [15, 19, 41, 45, 53] sem glavne dele, postopke in naloge DQM na zbiren način prikazal v sliki 6.



Slika 6: Elementi upravljanja kakovosti podatkov

DQM se najprej deli na dva dela: na proaktivni in reaktivni del. Reaktivni pristop se ukvarja z napakami, ki so zaradi različnih vzrokov, opisanih v točki 4, že prisotne v podatkovni bazi. Proaktivni pristop pa je namenjen zagotavljanju kakovosti prihodnjih podatkov.

3.4.2 Vloge DQM in skrbništvo podatkov

Vloge v DQM so naslednje [8, 15, 19]:

- vodja skrbnikov podatkov (ang. *chief data steward*),
- poslovni analitik (ang. *business analyst*),
- podatkovni analitik (ang. *data analyst*),
- tehnični skrbnik podatkov (ang. *technical data steward*),
- poslovni skrbnik podatkov (ang. *business data steward*),
- administrator kakovosti (ang. *quality administrator*),
- pokrovitelj (ang. *sponsor*),
- svet obvladovanja podatkov (ang. *data governance council*),
- lastnik podatkov (ang. *data owner*).

Geiger [19] opisuje nekatere vloge: **poslovni analitik** opiše poslovne zahteve, ki morajo vsebovati tudi podrobne zahteve glede kakovosti podatkov. **Podatkovni analitik** te zahteve prenese v podatkovni model, arhitekturo bodočega sistema ter v pravila ob postopku zajema in prenašanju podatkov. Ko podatkovni analitik naredi ustrezne načrte, jih posreduje razvijalcu za izdelavo programske opreme. Opisani postopek ponazarja slika 7. **Poslovni skrbnik podatkov** je odgovoren za upravljanje podatkov. Potrebuje tako tehnična znanja kot ustrezne osebnostne veščine. Tehnična znanja vključujejo:

- osnovno znanje podatkovnega modeliranja,
- osnovno znanje sistemov za upravljanje podatkovnih baz - DBMS,
- napredno znanje konceptov podatkovnih skladišč,
- sposobnosti tehničnega pisanja.

Osebnostne veščine in ostale potrebne lastnosti zajemajo:

- razumevanje poslovnega modela organizacije,
- organizacijske sposobnosti,
- komunikacijske sposobnosti,
- objektivnost,
- ustvarjalnost,
- diplomatske sposobnosti,
- sposobnost dela v skupini,
- uživanje zadostnega ugleda v organizaciji.

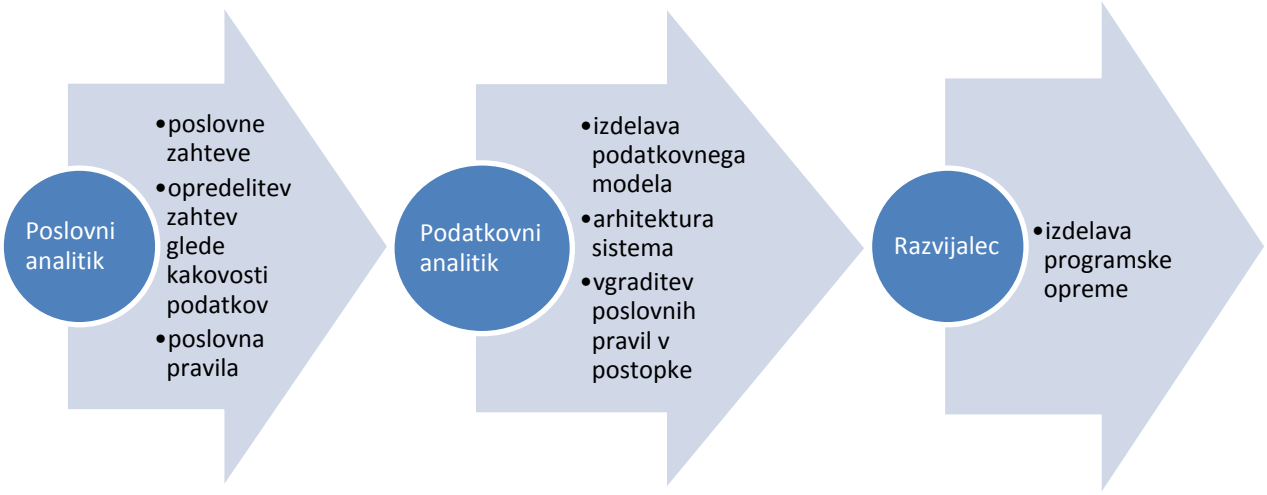
Falge, Otto in Österle [15] opisujejo preostale vloge: **vodja skrbnikov podatkov** se ukvarja s/z:

- usklajevanjem strateškim poslovnih ciljev s cilji DQM;
- jasno določitvijo obsega različnih področij podatkov, na katere vpliva strategija DQM;
- jasno določitvijo opravil DQM (nadzor, izvedba itd.);

- prikazom prispevka DQM celotni organizaciji;
- dolgoročnim načrtovanjem neprekinjenega vključevanja DQM v organizacijo;
- odvisnostjo DQM od drugih projektov.

Pokrovitelj spodbuja DQM v organizaciji in usmerja aktivnosti. **Svet obvladovanja podatkov**, ki ga sestavljajo **lastniki podatkov** in **vodja skrbnikov podatkov**, usklajujejo različne interese deležnikov funkcije DQM ter sprejemajo pomembnejše odločitve. **Lastniki podatkov** so odgovorni za pravilnost in skladnost določenih podatkov, medtem ko **skrbniki podatkov** pripravljajo pravila za delo s podatki. Kot potrjuje tudi Geiger [19], skrbništvo podatkov ni enako lastništvu.

Cappiello, Francalanci in Pernici [8] k navedenim vlogam dodajajo še vlogo **administratorja kakovosti**. Ta vloga je potrebna za upravljanje znanja o podatkovnih strukturah, operacijah in povezanih postopkih.



Slika 7: Sodelovanje nekaterih vlog

Odgovornosti skrbnika podatkov so obsežne, delijo se na več skupin, kot prikazuje slika 8.

Slika 8: Skupine odgovornosti skrbnika podatkov

Posamezne skupine zajemajo spodaj navedene odgovornosti skrbnika podatkov [19].
Odgovornosti **pridobitve podatkov** zajemajo:

- vzpostavitev poslovnih postopkov za izdelavo ali spremembo podatkov,
- vzpostavitev sistema za delo s podatki,
- vzpostavitev pooblastil za zajem in spremembo podatkov,
- vzpostavitev pravil potrjevanja podatkov,
- vzpostavitev poslovnih pravil za delo s podatki,
- vzpostavitev omejitev kakovosti podatkov oz. sprejemljive stopnje napak.

Odgovornosti **upravljanja podatkov** zajemajo:

- razvoj in vzdrževanje podatkovnega modela,
- razumevanje demografije podatkov,
- vzpostavitev standarda za poimenovanje objektov,
- izdelava zahtev za metapodatke in zagotovitev skladnosti podatkov z njimi,
- upravljanje redundantnosti podatkov,
- skrb za varnostno kopijo podatkov in reševanje podatkov,
- skrb za arhiviranje podatkov in obnavljanje podatkov.

Odgovornosti **širitve podatkov** zajemajo:

- definiranje varnostnih pravil dostopa in preverjanje skladnosti z njimi,
- izdelava standardnih poizvedb in poročil,
- zagotavljanje dostopa uporabnikom,
- upravljanje uporabe sistema,
- spremljanje kakovosti podatkov,
- zagotavljanje primernih metapodatkov.

Odgovornosti **uničenja podatkov** zajemajo:

- vzpostavitev pravil za hranjenje podatkov in preverjanje skladnosti z njimi,
- brisanje podatkov v skladu s poslovnimi pravili, zahtevami in zunanjimi direktivami.

Geiger [19] vidi vzpostavitev skrbništva oz. določitev članstva v tej vlogi kot enega izmed izzivov pri vzpostavljanju DQM, zato predlaga način za določanje skupin in članstva. Uporabimo matriko CRUD, ki predstavlja zastopanje skrbniških skupin, za vsako skupino podatkov, ki bodo predmet upravljanja DQM. Os X predstavlja področje podatkov, poslovne funkcije ali poslovna področja predstavlja os Y in je lahko bolj ali manj podrobno razdeljena. Na preseku obeh osi z vnosom črk prikažemo, ali so podatki določenega področja uporabljeni v določeni poslovni funkciji. Uporabimo črke C, R, U, D (od tod ime matrike), ki pomenijo:

- C – izdelava podatka (ang. *create*),
- R – branje podatka (ang. *read*),
- U – sprememba podatka (ang. *update*),
- D – brisanje podatka (ang. *delete*).

Za vsako področje podatkov (na osi X) mora biti vsaj eno izpolnjeno polje oz. skupina skrbnikov podatkov. Članstvo te skupine pa je naslednje: **lastniki procesa skrbništva** (lahko jih je torej več) morajo biti osebe iz poslovnih funkcij, označene s C, U ali D, torej tistih poslovnih funkcij, ki podatke ustvarjajo, spreminjajo ali brišejo. Člani te skupine morajo biti tudi osebe iz poslovnih funkcij, ki podatke berejo. Primer matrike, kjer so poslovna področja predstavljena na nivoju podrobnosti poslovnih funkcij [35]:

		Področje podatkov					
		Stranka	Izdelek	Naročilo	Pogodba	Zaposleni	Račun
Poslovne funkcije	Kadrovska f.					C, R, U, D	R
	Tehnična f.		R	R		R	
	Proizvajalna f.		C, R, U, D	R			
	Prodajna f.	C, R, U	R	C	R, U	U	C, R
	Finančna f.	R		U	C		R, U

Preglednica 1: Primer CRUD matrike

Glede na zgornjo preglednico 1 bi bile ustvarjene naslednje skupine:

- skupina skrbništva podatkov o strankah, kjer bi bil vodja nekdo iz prodajne funkcije, obvezni člani pa bi bili tudi iz finančne funkcije;
- skupina skrbništva podatkov o izdelkih, kjer bi bil vodja nekdo iz proizvajalne funkcije, obvezni člani pa bi bili tudi iz tehnične in prodajne funkcije;
- skupina skrbništva podatkov o naročilih, kjer bi bili vodji osebi iz prodajne in finančne funkcije, obvezni člani pa bi bili tudi iz tehnične in proizvajalne funkcije;
- skupina skrbništva podatkov o pogodbah, kjer bi bil vodja nekdo iz finančne funkcije;
- skupina skrbništva podatkov o zaposlenih, kjer bi bili vodji osebi iz prodajne in kadrovske funkcije, obvezni člani pa bi bili tudi iz tehnične funkcije;
- skupina skrbništva podatkov o računih, kjer bi bili vodji osebi iz prodajne in finančne funkcije, obvezni člani pa bi bili tudi iz kadrovske funkcije.

3.4.3 Izzivi vzpostavitve DQM

Vzpostavitev DQM v organizaciji predstavlja velik izziv, posebno zaradi sledeče problematike [19]:

- **Nobena organizacijska enota se ne čuti odgovorna za težave**. Ko so podatki shranjeni, se običajno problem kakovosti podatkov prenese na IT. Vendar IT ne more ustvariti poslovnih pravil. Kot že zapisano, IT le zagotovi, da so poslovna pravila razvita in da je delovanje pravilno. Težavo predstavlja tudi usmeritev pozornosti zaposlenih v poslovnem delu v področje dela s podatki ter določanje skrbnika podatkov.
- **Zahteva po sodelovanju med organizacijskimi enotami**. DQM ne zajema le aktivnosti v posameznih organizacijskih enotah ali funkcijah organizacije, ampak presega meje enot.
- **Organizacija se mora zavedati resnosti problema slabe kakovosti podatkov**. Pogosto se organizacije zavedajo, da imajo težava v kakovosti podatkov šele takrat, ko takšni podatki že povzročijo kakšno resno posledico.
- **Zahteva disciplino**. Odgovornosti in zadolžitve morajo izvajati vsi, ki so jim le-te dodeljene. Pri vnosu podatkov je treba poskrbeti za takšen nivo natančnosti posameznih podatkov, ki zadostuje vsem posameznim organizacijskim enotam organizacije. Nekateri oddelki imajo namreč različne potrebe po posameznih prvinah kakovosti. V oddelku za evidentiranje vloge stranke v informacijski sistem npr. ne potrebujemo popolnega naslova, ampak je dovolj nestrukturiran podatek, s pomočjo katerega stranko določimo. Oddelek, ki vlogo rešuje in pošlje stranki odgovor, pa potrebuje popoln in natančen podatek.
- **Zahteva finančna sredstva**. Vzpostavitev samega DQM jih sicer zahteva, po drugi strani pa jih zahtevajo tudi nekakovostni podatki, kot je navedeno v točki 2.
- **Zahteva človeške vire**. Po eni strani DQM zahteva vire za obvladovanje aktivnosti DQM, po drugi strani se lahko sprostijo viri na aktivnostih popravljanja napak v podatkih, ki se lahko nadomestijo z namenskimi programskimi rešitvami, ter viri z morebitnih projektov ali aktivnosti ročnega popravljanja podatkov.
- **Težko je izračunati povrnitev investicije**. V finančnem smislu je to oceno pogosto težko pridobiti. Lahko pa se ovrednoti negativni učinek, ki ga imajo nekateri znani problemi v podatkih, in tako upravičimo investicijo v DQM.

3.4.4 Vpeljava DQM

Geiger [19] za uspešno vpeljavo DQM predlaga, da se v začetku posvetimo predvsem naslednjim področjem:

- **Izobraževanje**. Izobraževanje je pomembno zaradi širšega sprejema in razumevanja DQM. Deležniki morajo poznati teoretična izhodišča, primere iz drugih organizacij ter konkretne težave v lastni organizaciji. Pridobitev podpore je lažja, če vodstvo spozna posledice težav v kakovosti podatkov.
- **Določitev skrbništva in odgovornosti.** Delitev odgovornosti mora biti naslednja; poslovni del organizacije skrbi za vzpostavitev poslovnih pravil za obvladovanje podatkov in za preverjanje in potrjevanje kakovosti podatkov. IT del organizacije pa skrbi za vzpostavitev in upravljanje okolja (arhitektura, tehnični pripomočki, sistemi in podatkovne baze), ki je zadolžen za pridobivanje, vzdrževanje, širjenje in uničenje podatkov.
- **Utrditev sodelovanja med deležniki**. Za uspeh upravljanja kakovosti podatkov je ključno sodelovanje med poslovnim in IT delom organizacije, za kar je potreben neoviran pretok informacij. Sodelovati morata tako poslovni in IT del organizacije kot tudi enote znotraj poslovnega dela in enote znotraj IT dela.
- **Štirifazni program.** Ta del je namenjen odpravi obstoječih napak v bazi – čiščenju podatkov. Vsebuje podobne korake kot Rahm in Do [45] in Chapman [9] in je naveden v točki 6.2.3.
- **Tehnološka podpora**. Tehnologija mora podpirati postopke DQM na takšen način, da lajša delo uporabnikom ter izboljšuje učinkovitost pri delu.

3.4.5 Ogrodje CDQM

V domeni DQM je bilo v preteklosti predstavljeno veliko pristopov in konceptov različnih avtorjev. Zbrali so jih Otto in ostali [41], ki so predstavili tudi svoje ogrodje Corporate Data Quality Management (CDQM):

- Total Data Quality Management (TDQM),
- Total Quality data Management (TQdM),
- Total Information Quality Management (TIQM),
- Framework for information quality management.

Ogrodje CDQM [41] vključuje pristope iz omenjenih predhodnih ogrodij z nekaterimi dopolnitvami (vključitev dolgoročnejše strateške usmeritve DQM ter določitev organizacijskih odgovornosti) ter iz najboljših praks ITIL (opisan v točki 5.1.2), ogrodja COBIT (opisan v točki 5.1.1) in Politik kakovosti podatkov (opisane v točki 5.1.5). Spodnja

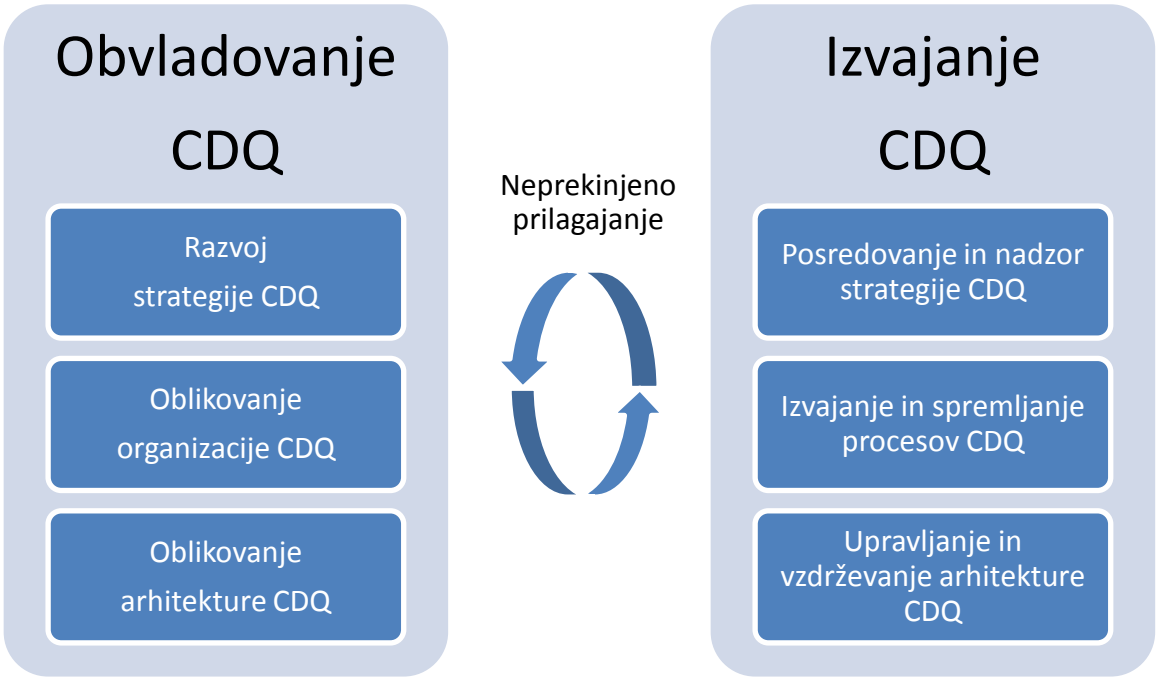
slika 9 predstavlja omenjeno ogrodje, ki je razdeljeno na dva vidika: obvladovanje in izvajanje.

Vidik obvladovanja se ukvarja z vprašanji:

- kaj je treba narediti;
- kdo mora kaj narediti;
- kako so porazdeljene odgovornosti.

Vidik izvajanja skrbi za izvedbo opravil.

Med omenjenima gradnikoma je kontrolna zanka, ki ponazarja odvisnost obeh vidikov. Posamezni elementi se po potrebi sproti prilagajajo.



Slika 9: Ogrodje CDQM

Ogrodje CDQM [41] je podrobneje opisano v nadaljevanju.

Vidik obvladovanja:

Razvoj strategije CDQ upravlja in usmerja vse aktivnosti, usklajeno s poslovno strategijo. Vključuje:

- razvoj strategije kakovosti podatkov, vključno s strateškimi cilji upravljanja podatkov;
- izdelavo ocene trenutnega stanja in vzpostavitev postopka ocenjevanja;

- določitev portfelja strateških pobud na podlagi ocene trenutnega stanja, s pomočjo katere prepoznamo kritične dele;
- oblikovanje poslovnega primera, ki materializira strateške pobude.

Razvoj strategije CDQM so avtorji podrobneje predstavili v poznejšem delu [15].

Oblikovanje organizacije CDQ vključuje:

- prepoznavanje informacijskih potreb uporabnikov [28];
- določitev postopkov nastajanja podatkov, vključno s kontrolami za zagotavljanje kakovosti, vpoglede, izdelavo itd.;
- določitev vlog in odgovornosti;
- določitev metrik kakovosti podatkov in standardov;
- vzpostavitev politik in postopkov, ki zagotavljajo nadzor, kakovost, upravljanje tveganj in varnost.

Oblikovanje arhitekture CDQ vključuje:

- razvoj skupnega modela informacijskih objektov, kar zajema podatkovni model, metapodatke, podatkovne sheme, stopnje varovanja, pravila in omejitve za vnos podatkov;
- izdelavo podatkovnega slovarja, ki temelji na metapodatkih, omenjenih v prejšnji točki in zbranih v podatkovnem slovarju BDD;
- določitev podpornega informacijskega sistema, kar zajema sistemsko arhitekturo, ki podpira upravljanje kakovosti podatkov.

Vidik izvajanja:

Posredovanje in nadzor strategije DQM vključuje:

- razvoj in izvrševanje načrta komuniciranja, Lee in ostali [28] priporočajo uporabo prijemov za motivacijo;
- izdelavo primernih kazalcev za spremljanje upravljanja organizacijskih sprememb;
- negovanje kulture učenja, ki sledi politiki nagrajevanja;
- spremljanje sprememb v okolju.

Izvajanje in spremljanje procesov DQM vključuje:

- spremljanje stopnje kakovosti podatkov z uporabo predhodno določenih metrik in kazalcev;
- spremljanje učinkovitosti in kakovosti postopka upravljanja podatkov;
- vpeljavo učinkovitega programa usposabljanja, ki zmanjšuje število napak, dviguje produktivnost in skladnost s kontrolami. Zaposleni, ki vnašajo

podatke, morajo razumeti, zakaj se podatki v naslednjih fazah poslovnih postopkov uporabijo.

Upravljanje in vzdrževanje arhitekture DQM vključuje:

- upravljanje in vzdrževanje sistema za shranjevanje in širitev podatkov;
- upravljanje in vzdrževanje sistema za analizo kakovosti podatkov, čiščenje podatkov, transformacijo podatkov, upravljanje metapodatkov in postopkov upravljanja podatkov.

3.5 Kakovost podatkov in področje interneta stvari

Internet stvari (ang. *Internet of Things*), v nadaljevanju IoT, je razmeroma mlado področje, ki je v zadnjem času deležno veliko raziskovalnega napora. Dela na tem področju so v svoji raziskavi med drugim analizirali Mishra, Lin in Chang [36]. V svoji raziskavi so predstavili problematiko kakovosti podatkov področja IoT ter ogrodje za upravljanje podatkov in pridobivanje znanja na področju avtomatizirane proizvodnje, kot bo navedeno v nadaljevanju.

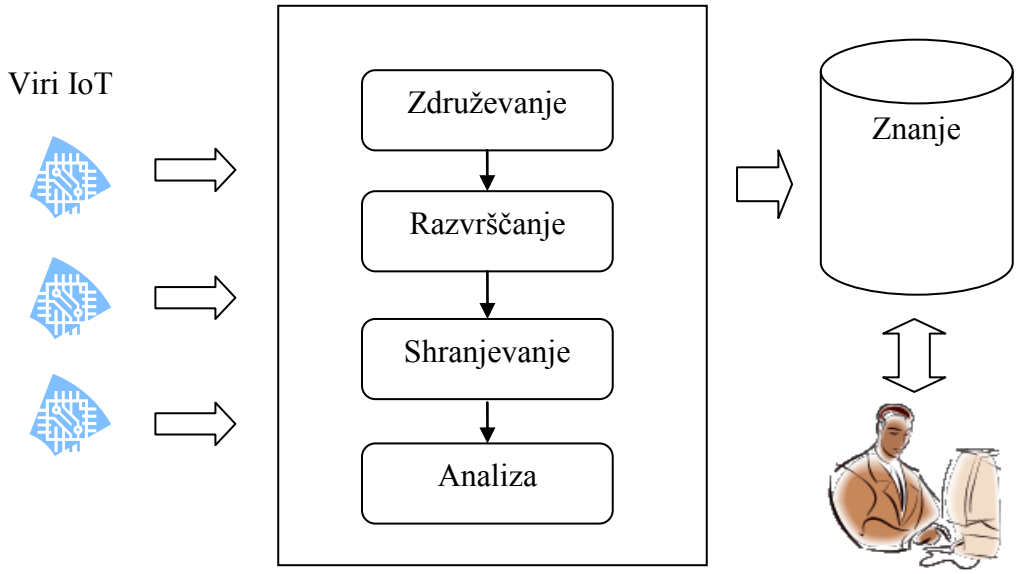
Objekti IoT so lahko kateri koli oprijemljivi predmeti v realnem svetu – ljudje, stroji, živali, naprave, ki imajo edinstveno identifikacijo in možnost dostopa do internetnega omrežja. Tehnično natančneje je IoT objekt senzor, RFID naprava ali druga naprava, ki ima povezavo po protokolu IP in je zmožna samostojnega pošiljanja podatkov brez posredovanja človeka [33]. IoT objekti se uporabljajo v širokem naboru področij: v finančnem, znanstvenem, industriji, zdravstvu, kmetijstvu, transportu itd. [36]

V okviru industrijske avtomatizacije obstaja na tisoče avtomatiziranih naprav, opremljenih z milijoni IoT čipi. Takšna mreža IoT objektov tvori velik obseg okolja IoT, ki generira ogromno količino strukturiranih, polstrukturiranih in nestrukturiranih podatkov v realnem času, ki jim pravimo tudi masovni podatki (ang. *big-data*) [2]. Podatki se nanašajo na temperaturo, svetlobo, vlažnost, pritisk, pospešek, hitrost, hrup, magnetno polje, koncentracije plinov itd. [57] Ti podatki se odložijo v zbirko podatkov, potrebni pa so za analize, modeliranje, pridobivanje znanja in odločanje. Količina podatkov, ki jih generira skupina IoT objektov, je navedena na primeru obsežnega električnega omrežja in znaša en terabajt na dan, upoštevajoč, da so podatki zajeti vsakih pet do petnajst minut [36].

Strukturirani podatki imajo točno določeno postavitev zapisa, medtem ko so polstrukturirani in nestrukturirani podatki v prosti obliki beleženja (t. i. log) in vsebujejo neenotna poimenovanja, merila, nivo abstrakcij itd. To povzroča redundanco, neskladnost, nepopolnost podatkov in ostale anomalije, predstavljene v opredelitvi pojmov, kar predstavlja težavo pri

obravnavi teh podatkov. Avtorji [36] so zato predstavili ogrodje COIB, predstavljeno na sliki 10, ki je sestavljeno iz več faz:

- **Faza združevanja masovnih podatkov**
  V tej fazi se izvršijo operacije združitve različnih tokov podatkov. Uporablja se standardna semantika za izključevanje anomalij kot tudi čiščenje podatkov. V rednih časovnih presledkih se podatki zajemajo iz različnih IoT virov ter se združujejo. Cilj te faze je, da se doseže celovitost, skladnost, popolnost in ostale prvine kakovosti podatkov, opisanih v točki 3.1.2. Kakovost podatkov se meri z ocenjevanjem. Potrebna je dovolj visoka ocena kakovosti za namen kakovostnih analiz podatkov v eni izmed prihodnjih faz. Področje ocenjevanja kakovosti je v literaturi tudi samostojno obravnavano, primer je delo Woodalla, Boreka in Parlikada [66] in Wattsa, Shankaranarayanana in Evena [62].
- **Faza razvrščanja masovnih podatkov**
  V tej fazi so podatki razvrščeni in ločeni v različne skupine glede na lastnosti podatkov in domeno, ki jo opisujejo (primer razvrščanja podatkov proizvodnje: operativni, proizvodni, vzdrževalni, statusni, inventarni itd.). Pri tem predstavlja potreba po obdelavi podatkov v realnem času poseben izziv.
- **Faza shranjevanja**
  V tej fazi se razvrščeni podatki shranijo v posamezne zbirke.
- **Faza analize masovnih podatkov**
  V tej fazi se podatki analizirajo, pri čemer se lahko uporabijo posebna analitična orodja.



Slika 10: Primer ogrodja za obdelavo podatkov v IoT

Osnovna arhitektura IoT je sestavljena iz več nivojev, različna literatura jo opisuje z različno stopnjo abstrakcije. Združen prikaz arhitekture raziskav [36, 57] je lahko naslednji:

<table>
<tr><td rowspan="2">Uporabniki aplikacij</td><td>Nivo aplikacij
(ang. application layer)</td></tr>
<tr><td>Nivo obdelave znanja
(ang. knowledge processing layer)</td></tr>
<tr><td>Platforma varnosti</td><td rowspan="3">Nivo vmesne programske opreme
(ang. middleware)</td></tr>
<tr><td>Storitve upravljanja naprav in komuniciranja</td></tr>
<tr><td>Platforma obdelave podatkov</td></tr>
<tr><td>Mreža</td><td>Nivo transporta
(ang. transport layer)</td></tr>
<tr><td>Naprave</td><td>Nivo fizične zaznave
(ang. physical sensing layer)</td></tr>
</table>

Preglednica 2: Arhitektura IoT in umeščenost upravljanja podatkov

Kot predlagajo avtorji, je opisano ogrodje COIB del vmesnega programja [36]. Platforma obdelave podatkov v zgornji arhitekturi je odgovorna za zgoraj naštete faze združevanja, razvrščanja in shranjevanja podatkov [57].

Čiščenje podatkov se izvršuje na nivoju vmesne programske opreme. De in ostali [13] so napravili pregled področja čiščenja masovnih podatkov ter predlagali metodo za popravek posameznih atributov v strukturirani podatkovni bazi. Obstoječe metode za čiščenje, ki jih omenjajo, so:

- odkrivanje odstopajočih vrednosti (ang. *outlier detection*),
- odstranitev šuma (ang. *noise removal*),
- razjasnitev entitet (ang. *entity resolution*),
- nadomeščanje manjkajočih vrednosti (ang. *imputation*),
- metoda, ki izkorišča zakonitosti oz. vzorce v podatkih, ki se izražajo v pogojnih funkcijskih odvisnostih CFD. Ta metoda je odvisna od dela čistih podatkov ali od zunanje tabele za učenje pravil, na podlagi katerih lahko čisti podatke.

Metoda, ki jo predlagajo, se imenuje BayesWipe. Posebnost te metode je ta, da za razliko od obstoječih ne potrebuje ekspertnega znanja s področja ali množice izvirnih čistih podatkov.

Povzamemo lahko, da glede na obravnavano literaturo [13, 36, 57] na področju IoT predstavljajo izziv glede kakovosti podatkov naslednja dejstva, predvsem pa njihova kombinacija:

- obseg podatkov,
- nestrukturiranost podatkov,
- semantični prepad med heterogenimi napravami,
- potreba po obdelavi podatkov v realnem času.

4. Vzroki slabe kakovosti podatkov

Glede na definicijo pojma kakovost podatkov moramo vzrok slabe kakovosti podatkov iskati v izvoru napak njenih posameznih komponent – npr. v celovitosti. Vzroki, ki v organizacijah pripeljejo do nepravilnosti v podatkih, so različni. Po pregledu literature lahko vzroke strnemo v naslednje skupine:

- arhitekturni vzroki,
- podedovani (zgodovinski) vzroki,
- organizacijski vzroki,
- varnostni vzroki.

4.1 Arhitekturni vzroki

Na eni strani gre za neustrezno načrtovanje že pri izgradnji – informatizaciji poslovnih procesov. Eden izmed virov neskladnosti v podatkih je arhitektura podatkovnega modela. Vzrok je lahko neuporaba ustreznih arhitekturnih pristopov, ogrodij in metodologij ali pa neustrezna (de)normalizacija, ki jo opisujejo Ling, Goh in Lee [32]. V njihovem delu pojasnjujejo teorijo normalnih oblik in primere uporabe, ki lahko ob neustrezni uporabi pripeljejo do težav – npr. uporaba tretje normalne oblike brez ustreznih dodatnih omejitev lahko povzroči neskladnost v primeru posodobitev podatkov. Rahm in Do [45] podobno menita, da je kakovost podatkov v veliki meri odvisna od ustrezne sheme in stopnje uporabe omejitev za kontrolo dovoljenih vrednosti v sami bazi kot viru. Do podobne ugotovitve je prišel Chapman [9], ki meni, da je morda najboljši način preprečevanja napak v pravilnem načrtovanju podatkovne baze.

Podatkovni model in posledično kakovost podatkov je tako odvisna tudi od znanja in izkušenj podatkovnega arhitekta. Oblikovanje dobrega podatkovnega modela je hkrati znanost in umetnost [32]. V preteklosti so že obstajali poskusi avtomatiziranja načrtovanja podatkovnega modela in poskusi izdelave modelov za razumevanje strokovnega znanja na tem področju, s čimer so se ukvarjali Storey, Thompson in Ram [54].

Rahm in Do [45] navajata številne vrste napak in jih uvrščata v različne skupine, kot je prikazano v točki 6.2.1. Kar pa je skupno vsem vrstam napak na nivoju zapisov, je to, da so posledica odsotnosti kontrol ob vnosu in odsotnosti omejitev na nivoju podatkovne baze.

Otto in ostali [41] navajajo dodaten vir napak. Napake pogosto nastanejo, kadar zbiramo podatke iz različnih poslovnih funkcij ali enot organizacije iz porazdeljenih sistemov.

Poseben primer predstavljajo masovni podatki, pridobljeni preko objektov IoT. Pri teh težava izvira iz same zasnove in je predvsem tehnične narave. Podrobneje je problematika prikazana v točki 3.5.

4.2 Podedovani (zgodovinski) vzroki

Težava se pojavi tudi ob prenovi informacijskega sistema. V preteklosti so bili različni sektorji v organizacijah informatizirani z različnimi informacijskimi sistemi (t. i. učinek silosov). Ob prenovi in enotnem informacijskem sistemu se zato ob selitvi podatkov iz podedovanih sistemov pojavi neskladnost podatkov, ker se kontrole v prejšnjih ločenih informacijskih sistemih niso uporabljale ali pa so se izvajale v ožjem obsegu. Temu pojavu se je težko izogniti, saj uporabniki v novem informacijskem sistemu pričakujejo in potrebujejo podatke, ki so jih uporabljali v starih sistemih [9, 19].

Boritz [6] navaja tudi dodaten zgodovinski vzrok pojavljanja napak v podatkih – kontrola podatkov se je v preteklosti izvajala večinoma le nad finančnimi podatki in informacijami, ne pa tudi nad podatki z drugih področij.

Chapman [9] navaja dodaten razlog: podatki so se pogosto zbirali priložnostno in ne sistematično [11, 65]. To še posebej velja za zbiranje podatkov v času, ki še ni bil podprt z IT. Primer, ki ga avtor obravnava, so določeni podatki s področja biologije, ki so se zbirali skozi 300 let.

4.3 Organizacijski vzroki

Boritz [6] navaja tudi nekoliko bolj posredne, netehnične vzroke. Kot vir navaja raziskavo Global data management survey iz leta 2001 [92], ki je prišla do zaključka, da je stanje v upravljanju podatkov slabo:

- dve tretjini odborov organizacij se z upravljanjem podatkov ne ukvarjata;
- dve tretjini odborov organizacij prenašata odgovornost za upravljanje podatkov izključno vodji IT (ang. *Chief Information Officer – CIO*) ali oddelku IT;
- polovica generalnih direktorjev (ang. *Chief Executive Officer – CEO*) upravljanja podatkov ne vidi kot strateško vprašanje;
- tretjina anketirancev je mnenja, da vodstvo ne posveča dovolj pozornosti upravljanju podatkov;
- le tretjina anketirancev je prepričana o kakovosti podatkov lastne organizacije;
- manj kot tretjina anketirancev je prepričana o kakovosti podatkov ostalih organizacij.

Geiger [19] navaja izsledek raziskave inštituta Data Warehousing, da povprečno 15-20 odstotkov podatkov v organizacijah vsebuje napake ali je kako drugače neuporabnih. Dodaja še, da so te težave v organizacijah pogosto spregledane in se jim ne posveča dovolj pozornosti.

Zgornji izsledki raziskave so skrb vzbujajoči, saj Gelbstein [20] ter Falge, Otto in Österle [15], trdijo, da je upravljanje s podatki odgovornost organizacije kot celote oz. da mora pobuda priti s strani poslovnih uporabnikov, IT pa je odgovoren za implementacijo. Gelbstein še izpostavlja, da je problematika prelaganja odgovornosti in lastništva podatkov v IT še posebej prisotna takrat, kadar IT storitve zagotavlja IT oddelek znotraj organizacije in ne zunanji izvajalec.

Suer in Nolan [55] navajata trditev, da je najmanj 20 odstotkov podatkov nepravilnih, ter oceno Bloor Reasearch, da lahko kvaliteta podatkov pade od 1 do 1.5 odstotka na mesec, če je ne upravljamo aktivno.

Gelbstein [20] navaja naslednje vzroke izgube celovitosti podatkov, kar posledično pomeni slabo kvaliteto podatkov. Vsem je skupno premajhno posvečanje pozornosti celovitosti:

- decentralizacija informacijskih sistemov;
- dostopnost razmeroma močnih programskih orodij za končne uporabnike, ki predstavljajo ranljivost. Kot primer navede preglednice, ki se uporabljajo za vodstvene odločitve. Takšne preglednice se pogosto uporabljajo za ročne vnose podatkov brez preverjanja vnesenih vrednosti;
- spremembe v pravilih za dostop in v pooblastilih;
- nezmožnost sledenja uporabe gesel z visokimi pooblastili;
- uporabniške napake, ki vplivajo na produkcijske podatke;
- ranljivosti v programski kodi aplikacij;
- neustrezen proces za spremljanje in odobritev sprememb;
- neustrezna nastavitev varnostnih naprav in varnostne programske opreme;
- neustrezno nameščanje popravkov programske opreme;
- vključevanje zunanjih naprav v notranje omrežje organizacij;
- neustrezna delitev odgovornosti;
- zlonamerna programska oprema;
- nepooblaščene spremembe nastavitev operacijskih sistemov.

Omenjeno drugo točko (potencialna ranljivost pri uporabi preglednic) potrjuje primer Fannie May, ki ga navaja Boritz [6].

4.4 Varnostni vzroki

Celovitost podatkov in varnost sistemov sta tesno povezana, kar potrjujejo številni avtorji, npr. [20, 21, 67].

Informacijska varnost je postala vidna težava v organizacijah [20]. Ogrodje Cobit v verziji 4.1 [82] v enem izmed svojih procesov Zagotovite varnost sistemov (DS5) navaja, da informacijska varnost med drugim izpolnjuje poslovno zahtevo vzdrževanja celovitosti informacij. To je skladno s trditvijo v [21], ki navaja, da varnostna strategija ne more uspeti brez zagotavljanja celovitosti podatkov. Poleg celovitosti pa sta komponenti informacijske ali podatkovne varnosti še razpoložljivost in tajnost [67]. Upravljanje podatkov, ki predstavljajo vir informacij, ogrodje Cobit [82] predlaga v procesu Upravljajte podatke (DS11), ki se nanaša na popolnost, pravilnost, razpoložljivost in varnost podatkov.

Celovitost podatkov se lahko ogrozi tako znotraj organizacije kot tudi od zunaj. Grožnje celovitosti, katerih vir je znotraj organizacij, so navedene že med organizacijskimi vzroki, saj menim, da ti dve skupini v tem delu sovpadata.

Gelbstein [20] navaja, da so podatkovne baze najmanj zaščiteni objekti v vsej IT infrastrukturi, kar je nedvomno neposreden vzrok težav v celovitosti in posledično slabe kakovosti podatkov. Kot možen vzrok temu navaja veliko različnih interpretacij in definicij podatkovne integritete (to lahko potrdimo s točko s pregledom definicij), ki se med seboj prekrivajo, naslavljajo različne probleme in tako ustvarjajo zmedo v pomenu. Do podobnega spoznanja glede števila različnih definicij, ki služijo posameznim specifičnim problemom, sta prišla že Zviran in Glezer [67].

Primer študije, ki se ukvarja s tem področjem ranljivosti organizacije od zunaj, sta izdelala Ferriyan in Istiyanto [17], ki sta obravnavala informacijsko varnost in prikazala ocenjevanje po zrelostnem modelu na primeru organizacije ter predlagala orodje za ocenjevanje ranljivosti in zrelostnega modela. Njuna študija se nanaša predvsem na mrežni nivo.

5. Pregled standardov in zakonodaje

5.1 Standardi in najboljše prakse

Standardi in najboljše prakse v IT so po [38] postale pomembne predvsem zaradi čedalje bolj konkurenčnega okolja, s čimer so se izoblikovali razlogi za njihovo vpeljavo:

- zahteva po boljši donosnosti investicij v IT,
- zahteva po povečanju poslovne vrednosti in zmanjšanju poslovnih tveganj,
- težnja vodstva po optimizaciji stroškov preko poenotenih pristopov,
- potreba organizacije po ocenjevanju poslovanja glede na splošno sprejete standarde – obstoj metrik.

Na področju upravljanja IT obstaja več standardov, zbirk najboljših praks. Vsi pregledani – [80, 81, 82, 85] – vsebujejo priporočila za upravljanje podatkov v IT, predlagajo pa nekoliko različne poudarke in nivo podrobnosti [38, 87]. ITIL in COBIT npr. ne nudita takšnega nivoja podrobnosti v varnosti kot ISO/IEC 27001. Po drugi strani COBIT nudi širok pregled nadzora in metrik IT, ITIL pa nudi podrobnejši nivo v procesih.

Vendar pa vpeljava teh standardov v organizacije ni samoumevna ali enostavna. Izzive, s katerimi se pri tem soočajo organizacije, opisuje [38] in so podrobneje pojasnjeni v točki 5.3, skupaj s koristmi pri vpeljavi.

Poleg zgoraj omenjenih standardov in najboljših praks pa obstajajo na tem področju tudi zakonodaja in direktive. Njihovo število raste, največ se jih nanaša na področje upravljanja s finančnimi podatki [20].

5.1.1 COBIT 5

COBIT [82] je ogrodje za obvladovanje IT in omogoča vodstvu premostitev vrzeli na področjih kontrol, tehničnih vprašanj in poslovnih tveganj. Omogoča razvoj jasnih politik in dobrih praks za kontrolo IT v organizacijah.

COBIT podpira obvladovanje IT z ogrodjem, ki zagotavlja, da:

- je IT usklajen s poslovnim delom organizacije;
- IT podpira poslovni del in potencira koristi;
- so IT viri porabljeni odgovorno;
- so IT tveganja primerno upravljana.

COBIT je usklajen z ostalimi standardi in se nenehno dopolnjuje, zaradi česar je postal okvir za vključevanje dobrih praks IT v organizacije in krovno ogrodje za obvladovanje IT. COBIT-ova procesna struktura omogoča popoln pregled nad IT na višjem nivoju, kar prinaša organizaciji največje koristi iz njenih investicij v IT [82].

Trenutna verzija ogrodja COBIT, izdana leta 2012, se imenuje COBIT 5. Predstavlja razširitev verzije 4.1 in vključuje ostala glavna ogrodja, standarde in vire, npr. ISACA Val IT, Risk IT, ITIL in sorodne standarde organizacije ISO [84]. Večja novost verzije 5 v primerjavi z verzijo 4.1 je še v razdelitvi organizacije na dve glavni področji, obvladovanje ali upravljanje na višjem nivoju (ang. *governance*) in upravljanje (ang. *management*) [83]:

- obvladovanje ali upravljanje na višjem nivoju je domena, ki vsebuje pet procesov; znotraj vsakega so definirane dejavnosti EDM: vrednotite (ang. *Evaluate*), usmerjajte (ang. *Direct*) in spremljajte (ang. *Monitor*);
- upravljanje je razdeljeno na štiri domene, vsaka pa vsebuje določeno število procesov. Omenjene štiri domene se skladajo z dejavnostmi PBRM: načrtujte (ang. *Plan*), izdelajte (ang. *Build*), poganjajte (ang. *Run*) in spremljajte (ang. *Monitor*).

COBIT ima v verziji 4.1 predviden samostojen proces Upravljajte podatke (DS11). Ta proces izpolnjuje poslovno zahtevo glede optimiziranja uporabe informacij in zagotavljanja razpoložljivosti informacij. Proces se tako usmerja na popolnost, pravilnost, razpoložljivost in varnost podatkov [82]. To je povezano s procesom Zagotovite varnost sistemov (DS5) – informacijska varnost med drugim izpolnjuje poslovno zahtevo vzdrževanja celovitosti informacij. Poleg celovitosti pa sta komponenti informacijske ali podatkovne varnosti še razpoložljivost in tajnost [67]. To pomeni, da je za uspešno upravljanje podatkov potreben tudi proces zagotavljanja varnosti sistemov.

V verziji COBIT 5 samostojni proces za upravljanje podatkov ni predviden. Kljub temu ima upravljanje podatkov v okviru COBIT 5 pomembno mesto. Avtorja v [55] pojasnjujeta umeščenost upravljanja podatkov v COBIT 5. Določila sta sedem aktivatorjev (ang. *enablers*), v vsakem so predvideni cilji in metrike za namen ustrezne kontrole in izboljšanja upravljanja in vodenja podatkov:

- upravljanje poslovnega tveganja, povezanega z IT,
- preglednost stroškov, prednosti in tveganj IT,
- varnost informacij, procesne infrastrukture in aplikacij,
- skladnost z notranjimi politikami,
- določitev in posredovanje sprejemljivega tveganja,
- učinkovito upravljanje kritičnih poslovnih tveganj, povezanih z IT,

- zagotovilo, da tveganja, povezana z IT, ne presežejo tveganja, sprejemljivega za organizacijo.

Hkrati pa lahko za posamezne vidike upravljanja podatkov uporabimo več posameznih procesov COBIT 5, npr. za varnostni vidik proces Upravljajte varnostne storitve (DSS5), za vidik kakovosti proces Upravljajte kakovost (APO11) itd.

5.1.2 ITIL 2011

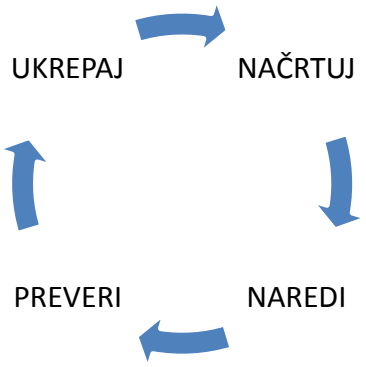
ITIL (Information Technology Infrastructure Library) [85] je zbirka najboljših praks na področju storitev IT in svetuje, kako uporabiti vire IT za lažje doseganje poslovnih koristi in rasti poslovanja. Metodologija ITIL podpira uskladitev storitev IT s poslovnimi procesi in poslovnimi potrebami. Procesi ITIL podpirajo poslovne potrebe organizacije s podporo njenih glavnih procesov. Trenutna verzija zbirke, izdana leta 2011, se imenuje ITIL 2011, kar je že četrta izdaja. Predhodne so bile ITIL, ITIL V2, ITIL V3.

Struktura in razdelitev ITIL V3 (struktura in razdelitev se v izdaji leta 2011 ni spremenila) je po [23], kot je navedeno spodaj.

Na najvišji ravni:
- jedrne publikacije (ang. *core publications*),
- dopolnilne usmeritve (ang. *complementary guidance*),
- spletne storitve (ang. *web support services*).

Jedrnih publikacij oz. področij je pet, vsaka pokriva del življenjskega cikla storitev:
- strategija storitev (ang. *service strategy*) – vsebina je priprava celovite strategije IT storitev v organizaciji;
- oblikovanje storitev (ang. *service design*) – vsebina je modeliranje, oblikovanje in zasnova IT storitev;
- prenos storitev (ang. *service transition*) – vsebina je implementacija storitev oz. prenos v produkcijsko okolje;
- izvajanje storitev (ang. *service operation*) – vsebina je izvajanje IT storitev v produkcijskem okolju;
- neprekinjeno izboljševanje storitev (ang. *continual service improvement*) – vsebina so procesi za izboljševanje kakovosti storitev in učenje posameznikov.

V osnovi ITIL procesi predstavljajo življenjski cikel, ki temelji na Demingovem krogu (slika 11) [39]. **Načrtuj** (ang. *plan*) zajema načrtovanje ali popravek poslovnega procesa za namen izboljšanja rezultatov. **Naredi** (ang. *do*) zajema izvedbo načrtovanega popravka in spremljanje izvajanja. **Preveri** (ang. *check*) zajema ocenjevanje meritev in posredovanje rezultatov nosilcem odločanja. **Ukrepaj** (ang. *act*) zajema odločanje o morebitnih popravkih.



Slika 11: Demingov krog

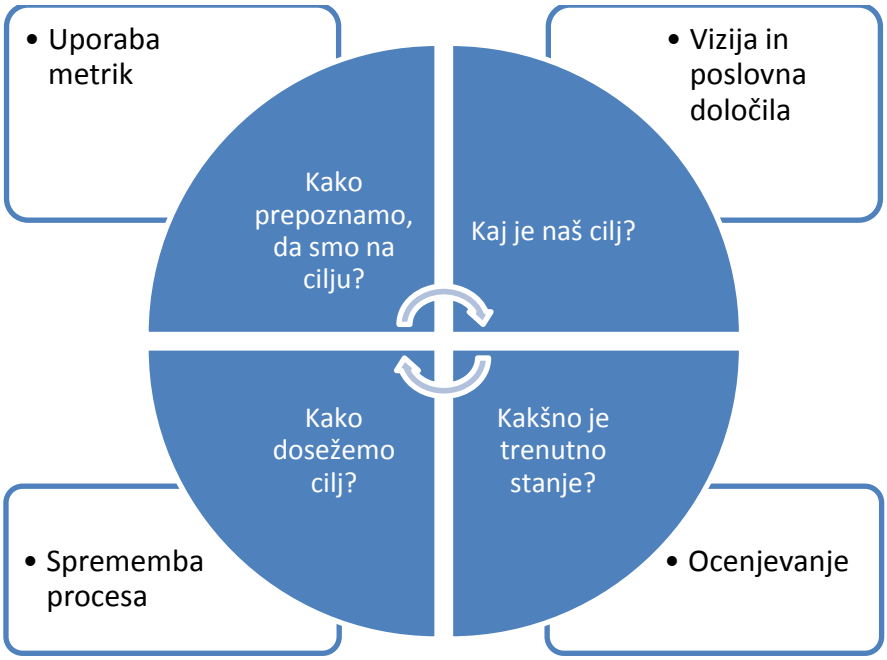
Avtor v [39] opisuje možno obravnavo kakovosti podatkov s pomočjo procesov ITIL, kot sledi v nadaljevanju.

Za uspešno vpeljavo programa kakovosti podatkov je v prvi vrsti potrebna podpora vodstva. Vzpostavljeno mora biti upravljanje podatkov na višji ravni (ang. *data governance*), nato naj bi sledila vzpostavitev skupine za nadzor kvalitete podatkov (ang. *data quality control board*). Usmerjala naj bi organizacijo v smislu želene stopnje kvalitete podatkov. Vzpostavljeno mora biti tudi lastništvo in dogovorjene odgovornosti. Člani skupine morajo biti tudi tiste osebe, ki imajo pooblastila za dodeljevanje finančnih sredstev in osebja in razumejo napor, ki ga terja zagotavljanje kvalitete podatkov.

Vzpostavljen program oz. njegovi člani morajo poznati odgovore na naslednja vprašanja:

- kakšna je želena stopnja kvalitete podatkov;
- kakšne je ocena trenutnega stanja;
- kako popraviti proces, ki je povzročil slabo kvaliteto podatkov;
- kako popraviti podatke v sistemu;
- kako ohraniti želeno stopnjo kvalitete podatkov.

Ta vprašanja so v osnovi skladajo z Demingovim krogom oz. osnovnimi vprašanji, ki se uporabljajo v področju neprekinjenega izboljševanja storitev (ang. *continual service improvement*), ki ga prikazuje slika 12.



Slika 12: Osnovna vprašanja področja neprekinjenega izboljševanja storitev

ITIL sicer ne pozna samostojnega procesa za namen zagotavljanja kakovosti podatkov, lahko pa uporabimo njegove obstoječe procese:

- podpora uporabnikom (ang. *service desk*) za javljanje problemov;
- upravljanje incidentov (ang. *incident management*) in upravljanje problemov (ang. problem management) za oceno trenutnega stanja kvalitete podatkov in določitev vzroka problema;
- upravljanje sprememb (ang. *change management*) in upravljanje verzij (ang. release management) preuči odkrite vzroke napak, razvije popravke glede na poslovne potrebe in jih namesti v produkcijsko okolje;
- upravljanje ravni storitev (ang. *service level management*) za spremljanje metrik in ukrepanje glede na doseganje dogovorjenih ravni storitev (ang. *Service Level Agreement – SLA*).

5.1.3 ISO/IEC 27001:2013 in ISO/IEC 27002:2013

Oba standarda sta del družine standardov ISO 27000, ki so namenjeni informacijski varnosti. ISO/IEC 27001:2013 [80] je standard, ki opisuje upravljanje sistema informacijske varnosti (ang. *Information Security Management System – ISMS*). Trenutna verzija je iz leta 2013. Predstavlja ogrodje načrtovanja in upravljanja informacijske varnosti, ne vsebuje pa natančnih

navodil za posamezne točke nadzora. Organizacije imajo tudi možnost certificiranja po tem standardu.

Standard ISO/IEC 27002:2013 [81] vsebuje navodila za izdelavo in upravljanje posameznih točk nadzora, definiranih v ISO/IEC 27001. Trenutna verzija je iz leta 2013. Vendar ISO/IEC 27002 ni standard, ki bi ga uporabljali samostojno za upravljanje sistema informacijske varnosti. Posledično tudi ni možno samostojno certificiranje po tem standardu [68].

5.1.4 DMBOK

DMBOK (The DAMA Guide to the Data Management Body of Knowledge) je zbirka najboljših praks in priporočil na področju upravljanja podatkov. Pri tem podobno kot COBIT ne vsebuje podrobnih tehničnih metod. Nekatera glavna področja, s katerimi se ukvarja DMBOK [72], so:

- upravljanje podatkov (ang. *data governance*),
- upravljanje podatkovne arhitekture (ang. *data architecture management*),
- upravljanje podatkovne varnosti (ang. *data security management*),
- podatkovno skladiščenje in poslovna inteligenca (ang. *data warehousing and business intelligence*),
- upravljanje z dokumenti in ostalimi vsebinami (ang. *document and content management*),
- upravljanje z metapodatki (ang. *meta-data management*),
- upravljanje kakovosti podatkov (ang. *data quality management*).

5.1.5 Data Quality Policy

Politike kakovosti podatkov iz knjige Journey to data Quality [28] predstavljajo deset smernic, ki naj bi bile osnova prizadevanjem za kakovost podatkov. Avtorji poudarjajo pomembnost kakovosti podatkov v organizaciji in jasno določitev vlog in odgovornosti. Pomembna smernica predstavlja koncept obravnave informacije kot produkta.

5.1.6 Payment Card Industry Data Security Standard (PCI DSS)

PCI DSS je standard, ki ga za upravljanje podatkov uporabljajo organizacije na področju kartičnega poslovanja [90]. Njegova trenutna verzija 3.1 je iz leta 2015.

5.1.7 ISO/ANSI SQL-89 in SQL-92

Standarda ISO/ANSI SQL-89 in SQL-92 sta standarda iz let 1989 in 1992 in sta med drugim uvedla uporabo referenčne integritete (SQL-89) oz. preverjanje celovitosti, kot jo opiše uporabnik preko kontrolnih omejitev (ang. *check constraints*) (SQL-92), s čimer se prispeva k celovitosti na nivoju podatkovne baze [67]. Ta uvedba je pomembna, ker se lahko z uporabo omenjenih omejitev že pred vnosom podatkov v podatkovno bazo izognemo nekaterim napakam v podatkih.

5.2 Zakonodaja

5.2.1 Zakon o varovanju osebnih podatkov (ZVOP)

Slovenski Zakon o varovanju osebnih podatkov (ZVOP-1-UPB1) iz leta 2007 [96], predvsem njegov 23. člen, upravljavcem zbirk osebnih podatkov nalaga zagotavljanje celovitosti podatkov s preprečevanjem nepooblaščenih dostopov do osebnih podatkov, varovanjem prostorov, opreme in programske opreme ter sledljivostjo sprememb osebnih podatkov.

5.2.2 Zakonodaja v tujini

Število zakonov in direktiv, ki se nanašajo na zagotavljanje informacij (ang. *information assurance*) in v tem okviru tudi na podatkovno celovitost, raste [20, 41]. Avtor v [20] omenja spodaj navedene zakone.

Zakoni v Združenih državah Amerike v izvirnem poimenovanju – za njihove kršitve pa so predvidene kazni – so:

- Data Quality Act,
- Sarbanes-Oxley Act,
- Gramm-Leach-Bliley Act,
- Health Insurance Portability and Accountability Act,
- Fair Credit Reporting Act,
- Federal Information Security Management Act.

Nekateri zakoni v Evropski Uniji v izvirnem poimenovanju pa so:

- Directive 95/46/EC (tudi Data Protection Directive) iz leta 1995;
- General Data Protection Regulation [76], ki bo nadomestil zakon iz prejšnje alineje;
- Council Framework Decision 2008/977/JHA iz leta 2008 [76];
- 8th Company Law Directive (Directive 2006/43/EC).

5.3 Izzivi in koristi pri vpeljavi standardov

Organizacije so k izvajanju zakonskih določil zavezane, ne pa tudi k upoštevanju standardov in najboljših praks. Prav zato si običajno poiščejo sebi primeren nabor standardov, ki jih upoštevajo. Kot opisujejo avtorji v [38], izbor in vpeljava standardov nista enostavna. Pri tem se namreč organizacije pogosto znajdejo pred nevarnostjo, da postane vpeljava draga in neučinkovita, predvsem v primerih, kadar se standarde jemlje le kot tehnično vodilo. Za učinkovito vpeljavo avtor priporoča predvsem naslednje predpostavke:

- osredotočenje na tiste dele poslovanja, kjer bi vpeljava standarda prinesla največ koristi;
- sodelovanje vseh nivojev poslovodstva – od najvišjega vodstva do posameznih vodij IT;
- zavedanje vodstva in osebja, zakaj je vpeljava pomembna;
- zavedanje vodstva in osebja, kakšne so pri tem njihove naloge in kako jih izvedejo.

Izbor standarda je po [38] odvisen od več dejavnikov:

- Proračunska sredstva, ki so na voljo. Vpeljava COBIT-a je običajno financirana iz proračuna organizacije, medtem ko je vpeljava ITIL-a in ISO/IEC 27002 financirana iz proračuna IT oddelka.
- Število IT procesov, ki jih želimo obvladovati. COBIT pokriva vse procese IT, medtem ko ITIL in ISO/IEC 27002 pokrivata specifične.
- Ali ima organizacija pregled nad vsemi procesi? Za vpeljavo COBIT-a organizacija potrebuje takšen pregled, medtem ko lahko ITIL vpeljujemo postopoma, proces za procesom.

Avtorji pri tem navajajo, da je vpeljava standardov in najboljših praks v organizacijo pomembna tako za boljše upravljanje IT, ki je kritično za uspeh strategije organizacije, kot tudi za obvladovanje aktivnosti IT. To pomeni, da ima organizacija tudi poslovne koristi – dvig učinkovitosti, zmanjšanje stroškov, manjše število napak in zaupanje poslovnih partnerjev. Predlagajo usklajeno uporabo več standardov – ogrodje COBIT na najvišji ravni za namen celovitega nadzora IT procesov (s tem se določi, ***kaj*** je treba narediti), specifične standarde (npr. ISO/IEC 27002), ogrodja ali najboljše prakse (npr. ITIL) pa na posameznih področjih (s tem se določi, ***kako*** je treba narediti), s čimer se ustvari hierarhija orodij za vodenje. Hkrati pa navajajo, da je vzpostavitev in sodelovanje več takšnih standardov izredno zapleten postopek in je bil že predmet več raziskav, katerih namen je bil doseči skladno medsebojno delovanje [22].

6. Obravnava slabe kakovosti podatkov

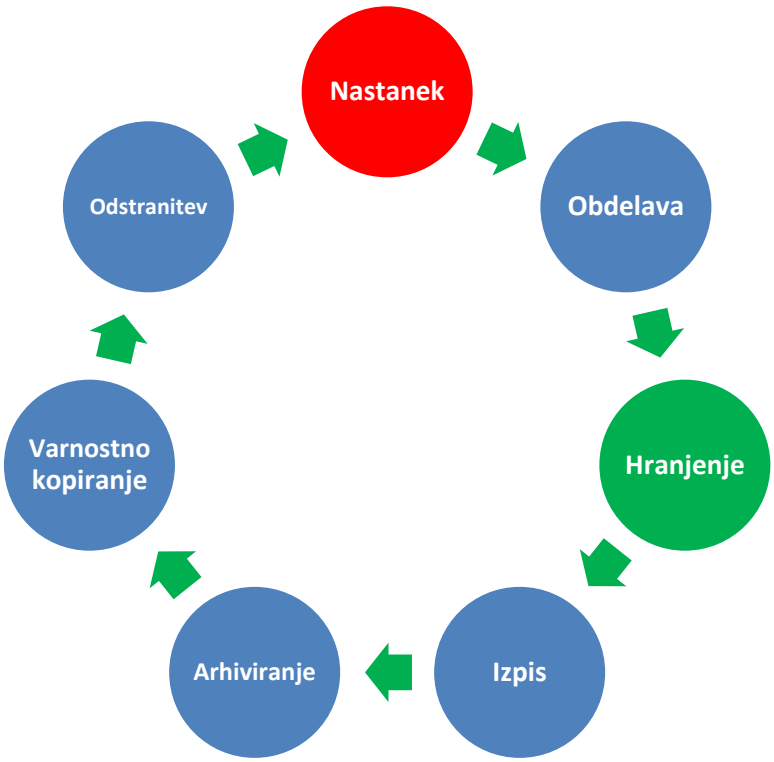
6.1 Najpogostejša mesta nastanka nepravilnosti

Življenjski cikel podatkov, kot ga predstavlja Gelbstein [20], je sestavljen iz naslednjih faz:

- vnos, izdelava ali pridobitev,
- obdelava,
- hranjenje, reproduciranje in razširjanje,
- arhiviranje in priklic,
- varnostno kopiranje in obnavljanje,
- izbris, odstranitev in uničenje.

Do nepravilnosti v podatkih lahko pride v kateri koli fazi življenjskega cikla. Storey, Dewan in Freimer [53] podobno navajajo z vidika poslovnih procesov – vsak korak v poslovnem procesu, od zajemanja podatkov do njihove uporabe v poslovnih odločitvah, ima vpliv na kakovost podatkov.

Slika 13 predstavlja življenjski cikel podatkov. Oblikovan je glede na faze, ki jih naštevata Boritz [6] in Gelbstein [20]. Zaradi preglednosti so poenotene in poenostavljene faze ter njihovo poimenovanje.



Slika 13: Življenjski cikel podatka

Avtor v [6] je izdelal raziskavo o najpogostejših mestih nepravilnosti oz. odstopanj od celovitosti. Raziskava je ocenjevala različne vidike sistema:

a) glede na pet faz obdelav podatkov:
   - vnos
   - prenos
   - obdelava
   - hranjenje
   - izpis

b) glede na pet faz sistemskega razvoja:
   - zasnova
   - načrtovanje
   - razvoj
   - uporaba
   - vzdrževanje

in c) glede na pet sistemskih komponent:
   - IT infrastruktura
   - programska oprema
   - človeški viri
   - postopki
   - podatki.

Omenjena raziskava je pokazala, da pri obdelavi podatkov največ nepravilnosti nastane ob vnosu (na zgornji sliki 13 obarvano rdeče), medtem ko v fazi hranjenja in prenosa nastane najmanj napak. Glede na pet faz sistemskega razvoja se je izkazalo, da največ napak nastane v uporabi sistema, medtem ko v fazi zasnove nastane najmanj nepravilnosti. Glede na pet komponent sistema najmanj nepravilnosti nastane zaradi IT infrastrukture, največ pa v postopkih.

6.2 Čiščenje podatkov

V literaturi obstaja več tujih izrazov za čiščenje podatkov [9, 45]:

- *data cleaning,*
- *data cleansing,*
- *data scrubbing,*
- *data validation,*
- *error checking,*
- *error detection,*
- *error correction.*

Področje čiščenja podatkov se ukvarja z odkrivanjem in odstranjevanjem napak v podatkih z namenom izboljšanja njihove kakovosti [10, 45].

Chapman [9] predlaga **uvrstitev čiščenja podatkov in hkrati preprečevanja napak v politiko upravljanja podatkov organizacije.** Čiščenje podatkov definira kot postopek za določitev netočnih, nepopolnih ali nerazumnih podatkov in izboljšanja kakovosti s popravkom odkritih napak in pomanjkljivosti.

Rahm in Do [45] ugotavljata, da to področje ni bilo deležno velike pozornosti v raziskovalni dejavnosti. To sta sicer zapisala že leta 2000, vendar se tudi do danes stanje ni bistveno spremenilo. Področje kot samostojna dejavnost ni pogosto predmet raziskav, večkrat je vključeno v okvir področja upravljanja kakovosti podatkov, npr. [19], ali pa so raziskane specifične metode za namen čiščenja podatkov, npr. tehnika "združi in očisti" (ang. *merge/purge*) [26]. Kot navajata, je čiščenje podatkov pomembno in se lahko uporabi v več primerih:

- v podatkovnih skladiščih kot glavni del procesa ETL;
- v primeru informacijskih sistemov, temelječih na spletnih tehnologijah;
- v primeru podatkov, temelječih na XML;
- v primeru posameznih zbirk podatkov (npr. centralna relacijska podatkovna baza IS).

K scenarijem uporabe, ki jih v predhodnem seznamu navajata Rahm in Do [45], bi lahko dodali še uporabo v primeru združitve različnih podatkovnih virov v enotno podatkovno zbirko (npr. selitev podatkov iz podedovanih sistemov v novo produkcijsko okolje). Po mojem mnenju je to pomemben in primeren scenarij za uporabo čiščenja podatkov. Zaposleni v IT namreč pri svojem delu namreč pogosto naletimo na napake, ki so posledica omenjenega scenarija.

Da so podedovani podatki posebno primerni za čiščenje podatkov, sta zapisala tudi Chapman [9], ki se je ukvarjal s podatki muzejev, zbranih v zadnjih 300 letih na področju biologije, ter Geiger [19].

6.2.1 Vrste napak v podatkih

V literaturi obstaja več možnih klasifikacij napak, spodaj sta navedeni dve. Chapman [9] je razvrščanje napak opravil za namen čiščenja podatkov v domeni biologije. Kljub temu se njegova razvrstitev v veliki meri prekriva z bolj splošno, širše uporabno razvrstitvijo, ki sta jo predlagala Rahm in Do [45].

Chapman [9] deli napake glede na vir podatkov:

- en sam vir,
- več virov (v primeru združevanja zbirk).

Na nivoju posameznih zapisov pa avtor razvršča napake glede na vrsto podatka:

- prostorske napake (ang. *spatial error*). Te napake se nanašajo na nepravilno geografsko lokacijo. Avtor je razdelitev napak izdelal za poseben namen v domeni biologije, kjer so prostorski podatki pomembni, zato jih je uvrstil v posebno skupino napak. Literatura prostorske podatke in njihovo kakovost tudi posebej obravnava, primer je delo Lia, Zhanga in Wua [31].
- izrazne ali sintaktične napake poimenovanja (ang. *nomenclatural error*). Te napake pomenijo napačna črkovanja imen in podobno. Za popravek teh napak se lahko uporabi različne slovarje in šifrante.
- pomenske, semantične ali napake taksonomije (ang. *taxonomic error*). Te napake pomenijo nepravilno uvrstitev ali pomensko nepravilno identifikacijo objekta. Izraz je v tem primeru lahko sintaktično pravilen, vendar je izbran popolnoma napačen izraz. Odkritje in odprava teh napak sta najtežja.
- opisne napake (ang. *descriptive error*). To so različne napake na preostalih, opisnih podatkih.

  Posamezne vrste napak, naštete zgoraj, avtor deli na:
  - manjkajoče vrednosti,
  - nepravilne vrednosti,
  - več podatkov v enem atributu,
  - vrednost v napačnem atributu,
  - redundanca – podvojenost zapisov,
  - napake skladnosti redundantnih zapisov,
  - napake skladnosti redundantnih podatkov.

V primeru več virov se pojavljajo enake napake kot v primeru enega vira, z dodatnimi pojavitvami podvojenosti zapisov. Avtor za takšne primere predlaga le primerno označitev zapisov in ne brisanja enega od njiju. V tem se razlikuje od pristopa, ki ga opisujeta naslednja avtorja.

Klasifikacijo napak v podatkih in opise težav posameznih skupin, ki sledi v nadaljevanju točke 6.2.1, sta izdelala Rahm in Do [45].

Avtorja napake najprej delita glede na vir podatkov:

- en sam vir,
- več virov.

Nadalje se obe vrsti delita glede na nivo pojavitve napake:

  - nivo sheme,
  - nivo zapisa.

Pri tem posamezne vrste napak niso neodvisne, pač pa so povezane. Napake enega vira se odražajo tudi v napakah več virov, prav tako se napake na nivoju sheme odražajo v posameznih zapisih.

Napake na **nivoju sheme enega vira** nastanejo zaradi:

- pomanjkanja primernega nabora omejitev (ang. *integrity constraints*) podatkovnega modela,
- slabo oblikovanega podatkovnega modela ali sheme.

Te napake so naslednje:

- nedovoljene vrednosti,
- kršena odvisnost atributov,
- kršena edinstvenost,
- kršena referenčna celovitost.

Napake na **nivoju zapisov enega vira** pa so tiste napake, ki jih ne moremo preprečiti na nivoju sheme:

- manjkajoče vrednosti,
- napake črkovanja oz. napačne vrednosti,
- okrajšave,
- več podatkov v enem atributu,
- vrednost v napačnem atributu,
- nasprotujoče si vrednosti v dveh atributih,
- redundanca – podvojenost zapisov,
- napake skladnosti redundantnih zapisov,
- napake skladnosti redundantnih podatkov,
- napačne vrednosti.

Napake obeh nivojev se lahko združujejo v naslednje skupine:

- napake v okviru atributov,
- napake v okviru zapisa,
- napake v okviru atributov,
- napake na izvoru.

Napake enega samega vira se pri združevanju virov stopnjujejo. Posamezni viri so razviti za specifične namene in se tako razlikujejo v shemah, naboru možnih vrednosti, pomenu posameznih vrednosti, naboru atributov itd. Napake zaradi več virov nastanejo pri običajni obravnavi v podatkovnih skladiščih, poleg tega pa tudi na primer pri selitvi podatkov iz podedovanih sistemov v nov informacijski sistem, na kar je opozoril tudi Geiger [19].

Glavne težave na **nivoju sheme več virov** (razlike med viroma) so [3, 24, 42]:

- razlike v poimenovanju (rešujejo se s preimenovanjem), le-te se delijo na:
    - isto ime se uporablja za različni podatkovni objekt;
    - različno ime se uporablja za isti podatkovni objekt;
- razlike v strukturi (zahtevajo prestrukturiranje in združevanje shem), primeri le-teh so:
    - isti objekt v enem viru opredeljen s tabelo, v drugem pa le z atributom;
    - različna struktura posameznih objektov;
    - različni podatkovni tipi;
    - različne omejitve celovitosti
    - itd.

Težave **na nivoju zapisov več virov** so vse tiste, ki so bile navedene že na nivoju zapisov enega vira. Pojavijo pa se dodatne. Tudi ko so zapisi v vsakem viru posebej brez napak, lahko pride do težav pri združevanju virov:

- različne predstavitve vrednosti,
- različen pomen vrednosti,
- različna stopnja agregacije,
- različni časi, na katere se podatki nanašajo
- itd.

Največji problem za čiščenje podatkov iz različnih virov je prepoznavanje istega fizičnega objekta, ki ga zapisi iz različnih virov opisujejo na različne načine (t. i. problem *merge/purge*). Pogosto so podatki iz različnih virov le delno redundantni, delno pa se dopolnjujejo. Cilj čiščenja podatkov v tem delu je, da združi redundantne podatke, dodatne podatke iz zapisov različnih virov pa doda v skupni, enotni zapis.

6.2.2 Vodila in smernice čiščenja podatkov

Chapman [9] navaja naslednja vodila:

- **Načrtovanje (razvoj vizije, politike in strategije).** Avtor predlaga uvrstitev čiščenja podatkov v vizijo in politiko kakovosti podatkov organizacije. Strategija vključitve čiščenja podatkov v kulturo organizacije bo kot rezultat izboljšalo kakovost podatkov in povečalo ugled organizacije.
- **Primerna organizacija podatkov izboljšuje učinkovitost.** Pred izvedbo analize podatkov je priporočljivo podatke najprej ustrezno organizirati – glede na različne kriterije, odvisno od primera uporabe. Če smo npr. gotovi, da so v določenem naboru podatkov le slovenski naslovi, je iskanje napak lahko enostavnejše in učinkovitejše, kot če bi uporabljali na vseh zapisih enotno logiko odkrivanja napak.
- **Preprečevanje je boljše kot kasnejše čiščenje.** Kot že omenjeno v točki 3.3, je za organizacijo preprečevanje napak v podatkih cenejše kot čiščenje. Pomembno je, da se mehanizmi preprečevanja napak dopolnjujejo. Ko s čiščenjem odpravimo napake, moramo poskrbeti za ustrezno dopolnitev mehanizma za preprečevanje napak.
- **Primerna delitev odgovornosti.** Vsak uporabnik lahko pomembno prispeva h kakovosti podatkov, med njimi pa se mora vzpostaviti primeren nivo sodelovanja.
- **Sodelovanje izboljšuje učinkovitost.** Če uspemo vzpostaviti visok nivo sodelovanja, je večja verjetnost, da nam uporabniki sporočijo najdene napake. Manj bo tudi ponovljenih analiz podatkov, napake bodo z večjo verjetnostjo dokumentirane in odpravljene in manj bo nastajanja novih napak zaradi neustreznega popravljanja.
- **Primerna postavitev prioritet.** Ustrezne prioritete znižujejo stroške in izboljšujejo učinkovitost. Pogosto se najprej osredotočimo na tiste zapise, ki jih lahko uredimo najprej ter z najmanjšimi stroški. Običajno je to tisti del podatkov, ki jih lahko uredimo programsko s paketnimi obdelavami.
- **Postavitev ciljev in metrik.** Cilji in metrike so namenjeni za uspešno upravljanje postopka čiščenja podatkov.
- **Izogibanje podvojenemu čiščenju.** Za namen ponovne obravnave podatkov, ki so že bili v postopku čiščenja, avtor predlaga ustrezno načrtovanje podatkovnega modela – npr. zagotovitev ustreznih atributov za vodenje podatkov o zapisih:
    - katere kontrole so se izvedle;
    - kakšen je bil rezultat kontrol;
    - kateri popravki podatkov so se izvedli;
    - kdo je izvedel omenjeni aktivnosti;
    - kdaj sta bili aktivnosti izvedeni.
- **Zagotavljanje povratnih informacij.** Končni uporabniki pogosto lažje zaznajo določene posamezne napake, ker podatke uporabljajo in jih lahko primerjajo z različnimi viri. Pomembno je, da so uporabniki obveščeni o sprejemu in popravku

podatkov. Predlagan je standarden mehanizem za zagotavljanje takšnih informacij. Na ta način se izboljšuje tudi sodelovanje.

- **Izobraževanje.** Pomembno je izobraževanje uporabnikov, ki podatke vnašajo ali imajo možnost njihovih sprememb, saj le-to zmanjšuje stopnjo napak iz tega vira.
- **Zadolžitve, preglednost, sledljivost spremembam.** Poskušamo se izogibati *ad hoc* pristopu k čiščenju. Namesto tega mora biti v politiki in strategiji razvidno, kdo je za kaj zadolžen ter na kakšen način se aktivnosti izvajajo. Sledljivost zagotovimo z vodenjem podobnih podatkov kot za izogibanje podvojenem čiščenju.
- **Dokumentiranje**. Dokumentacijo moramo voditi na nivoju zbirke podatkov ter na nivoju vsakega zapisa. Za vsak zapis moramo voditi podobne podatke kot pri izogibanju podvojenem čiščenju, našteti so zgoraj.

Kot navedeno že v točki 3.3, Gelbstein [20] za izboljšanje stanja na področju celovitosti podatkov navaja t. i. pravilo:

- dveh D (Detect, Deter),
- dveh P (Prevent, Prepare),
- dveh R (Respond, Recover)

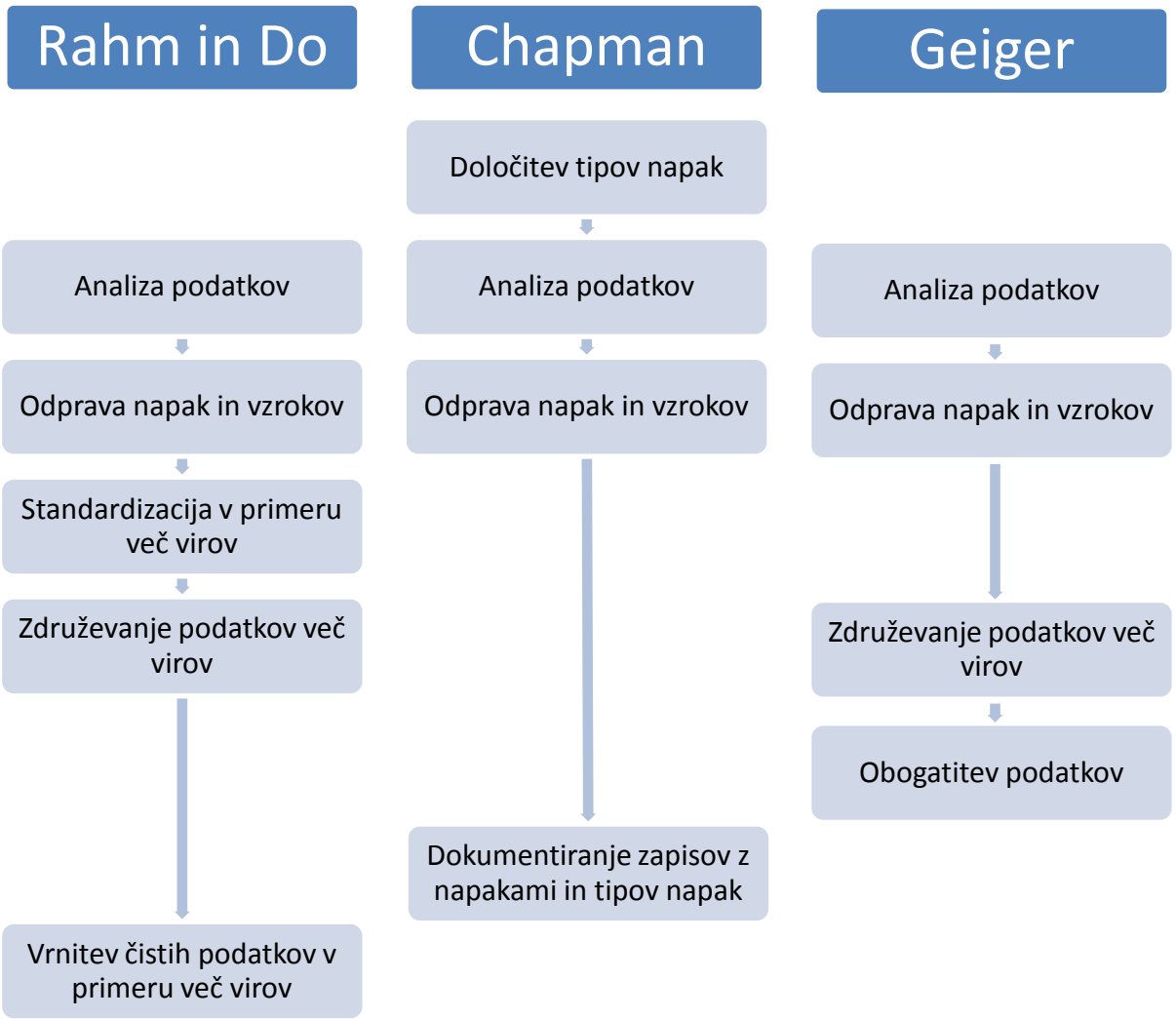
ali "odkrij, odvrni, preprečuj, pripravi, reagiraj, povrni". Koraki za zagotavljanje celovitosti podatkov naj bi naslavljali ta pravila.

Ker je celovitost ena izmed dimenzij kakovosti podatkov (glej točko 3.1.2), bi ta pravila morali upoštevati tudi v pristopu k čiščenju podatkov. V tem kontekstu bi lahko omenjena pravila razumeli in realizirali na naslednji način:

- **odkrij**: izvedba analize podatkov;
- **odvrni** in **preprečuj**: odprava nepravilnosti v postopkih in programski opremi, ki napake povzroča, ter vgradnja dodatnih omejitev za preprečevanje napak;
- **pripravi**: priprava ustreznih postopkov, vlog, odgovornosti in dolžnosti na nivoju organizacije;
- **reagiraj** in **povrni**: izvedba čiščenja podatkov ter obveščanje virov napak.

6.2.3 Postopki čiščenja podatkov

V tej točki bo predstavljenih nekaj ogrodij za čiščenje podatkov, kot so predlagani v literaturi. Na sliki 14 so prikazani glavni poudarki predstavljenih ogrodij [19, 34, 45]. Z nje je razvidno, da je jedro postopka v vseh treh ogrodjih enako (analiza podatkov, kateri sledi odprava napak in vzrokov). Razlikujejo pa se v posameznih dodatnih korakih, ki jim nekateri avtorji dajejo poudarek, drugi pa ne. Združeni pogled sem izdelal v točki 6.2.4.



Slika 14: Poudarki različnih postopkov

Chapman [9] za čiščenje podatkov predlaga splošno ogrodje, ki sta ga definirala Maletic in Marcus [34]. Vsebuje naslednje zaporedje faz:

- definicija in določitev tipov napak,
- iskanje in prepoznavanje zapisov z napakami,
- odstranjevanje napak,
- dokumentiranje zapisov z napakami in tipov napak,
- odstranitev vzroka napak, kadar je to možno.

Geiger [19] čiščenje podatkov umešča v širši okvir DQM. Faze znotraj čiščenja označuje kot štiri stebre DQM. Te opisuje kot:

- Opisovanje podatkov (ang. *data profiling*). V tem koraku gre za analiziranje podatkov po posameznih atributih in primerjavo s specifikacijo kakovosti.
- Odprava napak. S pomočjo rezultatov prejšnjega koraka poskušamo odkriti vzroke napak ter vzroke napak odstraniti s spremembo ustreznih postopkov in programske opreme. Odkrite obstoječe napake pa obravnavamo na naslednje možne načine:
    - Izključitev podatka. Če je napaka velika in je ni mogoče popraviti, takrat je to sprejemljiva izbira.
    - Sprejem podatka. To možnost izberemo, kadar je napaka v okviru dovoljenih odstopanj, kot jih je določil skrbnik podatkov.
    - Popravek podatka. To možnost uporabimo, kadar imamo na voljo pravilen podatek.
    - Zamenjava s privzeto vrednostjo. To možnost je smiselno uporabiti, kadar vrednost za določen atribut mora obstajati, nimamo pa na voljo pravilne vrednosti.
- Združevanje podatkov (ang. *data integration*). Ta korak uporabimo, kadar združujemo podatke iz več podatkovnih virov. Glavni aktivnosti v tem koraku sta prepoznavanje istega objekta v več podatkovnih virih (ang. linking) in združitev podatkov iz vseh obravnavanih podatkovnih virov (ang. *consolidation*).
- Obogatitev podatkov (ang. *data augmentation*). Obogatitev podatkov je korak za oplemenitenje podatkov organizacije s podatki iz zunanjih virov. S tem je podatkovni zbirki organizacije dodana nova vrednost, saj lahko organizacije na podlagi skupnih podatkov izdelajo analize in tako pridejo do novih informacij. Organizacije to pogosto izkoriščajo za razvrščanje strank in usmerjeno trženje.

Rahm in Do [45] menita, da mora pristop ali postopek za čiščenje podatkov zadostovati naslednjim zahtevam:

- zaznavanje napak na enem ali več virih,
- odstranjevanje napak,
- podprtost z orodjem za namen izogibanja ročnih pregledov podatkov,
- razširljivost za namen vključitve dodatnih virov,
- poleg čiščenja podatkov se mora ukvarjati še z dopolnjevanjem obstoječih shem in programske opreme,
- obstoj transformacijskih pravil za namen uporabe v dodatnih virih in v izvajanju poizvedb.

Za čiščenje podatkov nato predlagata in opisujeta naslednje faze:

Analiza podatkov

V tej fazi odkrijemo vrste napak, ki so prisotne. Uporablja se ročni pregled podatkov (poizvedbe), priporočljiva pa je tudi uporaba namenskih programov za pridobitev metapodatkov o lastnostih podatkov. Ti lahko prispevajo pri iskanju ujemanja atributov med posameznimi shemami. S preverjanjem zapisov se pridobi tudi neobičajne vzorce v vrednostih podatkov. Rezultate analize je priporočljivo shraniti za namen ponovne uporabe. Analizo podatkov delimo na:

- opisovanje podatkov (ang. *data profiling*) in
- podatkovno rudarjenje (ang. *data mining*).

Opisovanje podatkov analizira posamezne zapise po atributih (podatkovni tipi, dolžina, nabor vrednosti, razpon vrednosti, pogostost posameznih vrednosti, vzorci v besedilu, edinstvenost, pojavitve vrednosti *null* itd).

Podatkovno rudarjenje pa išče zakonitosti in vzorce med več atributi na večjih množicah podatkov.

Definicija transformacijskega toka in pravil

Ta faza vključuje določeno število transformacij podatkov in korakov čiščenja. Število je odvisno od stopnje raznolikosti in čistosti podatkov. Kadar je potrebno sheme več virov prevesti v skupno shemo, se uporabi prevajalna shema.

Začetni koraki se uporabijo za čiščenje napak zapisov enega vira. S tem se podatki pripravijo za združevanje v skupni model. Ti koraki vključujejo:

- Določitev posameznih vrednosti iz prostih besedilnih nizov. Ta korak je potreben, da dobimo posamezne vrednosti v ločenih atributih, s katerimi lahko delamo v nadaljevanju postopka.
- Potrjevanje in popravljanje. Korak vključuje pregled vsakega zapisa vira, iskanje in popravek napak, v kolikor je ta mogoč. Uporablja se preverjanje besed po različnih slovarjih (tako se lahko odpravi napake v črkovanju, elemente naslova itd.), odvisnosti atributov in ostale vrste napak zapisov enega vira.

  Klasične in ostale metode, ki se lahko uporabijo [13]:
  - odkrivanje odstopajočih vrednosti (ang. *outlier detection*),
  - odstranitev šuma (ang. *noise removal*),
  - razjasnitev entitet (ang. *entity resolution*),
  - nadomeščanje manjkajočih vrednosti (ang. *imputation*),
  - metode za čiščenje podatkov na osnovi učenja zakonitosti v podatkih, vendar pa za ta namen potrebujemo del čistih podatkov, na podlagi katerih algoritem določi pravila. Odsvetuje pa se učenje pravil neposredno na umazanih podatkih, saj so rezultati čiščenja v tem primeru slabi.
  - metoda BayesWipe, ki za čiščenje uporabi pravila, naučena iz prvotne množice umazanih podatkov. Podrobno jo opisuje [13].
- Standardizacija. Standardizacija je potrebna zaradi kasnejšega lažjega ujemanja zapisov iz več virov in združevanja. Zato morajo biti posamezni atributi posameznih virov poenoteni po podatkovnem tipu in obliki (npr. datumska oblika zapisa mora biti enotna, odstranjeni morajo biti prazni nizi pred ali za besedami itd.).

Poznejši koraki zajemajo združevanje shem in podatkov in čiščenje napak na zapisih več virov. Najprej se uredijo napake na nivoju sheme, kar zajema delitev, združevanje, razširitev, skrčenje atributov in tabel. Nato se uredijo napake na nivoju zapisov. V tem sklopu je zadnja aktivnost izločitev podvojenih zapisov. Izvede se na obeh virih ali pa na že združeni množici. V ta namen je treba najprej prepoznati, kateri zapisi iz različnih virov sodijo skupaj. V najboljšem primeru imamo na voljo identifikator, ki enolično določa par. V nasprotnem primeru moramo uporabiti ustrezno tehniko, znano kot "združi in očisti" (ang. *merge/purge*). Uporabo te tehnike podrobneje obravnava [26].

Za definiranje pravil transformacije in čiščenje na nivoju sheme avtorja priporočata uporabo deklarativnega programiranja, npr. poizvedb SQL ali uporabo grafičnega uporabniškega vmesnika, zaradi samodejnega ustvarjanja transformacijske kode. Uporabne so tudi uporabniško definirane funkcije UDF. Vanje je lahko vgrajena posebna logika transformacij, ista funkcija se lahko uporabi na več mestih in skrajšuje čas dostopa do podatkov. Primer takšne funkcije je lahko npr. določanje posameznih elementov naslova iz enotnega besedilnega niza o naslovu.

Želena je tudi možnost vstavljanja in proženja kode, ki jo je ustvaril uporabnik oz. izvajalec, ter dodatnih orodij za primer posebnih potreb transformacij. Transformacijski koraki lahko zahtevajo odziv in odločitev uporabnika v primerih zapisov, za katere ne obstaja vgrajena logika čiščenja.

Preverjanje

Ta faza služi za testiranje pravilnosti in učinkovitosti transformacij. To lahko naredimo na manjši množici testnih podatkov ali na kopiji izvirne množice podatkov. Po potrebi prilagodimo pravila. Pogosto je potrebnih nekaj ponovitev prilagoditev pravil in ponovnega preverjanja, ker se nekatere napake izrazijo šele po predhodnih transformacijah.

Transformacija

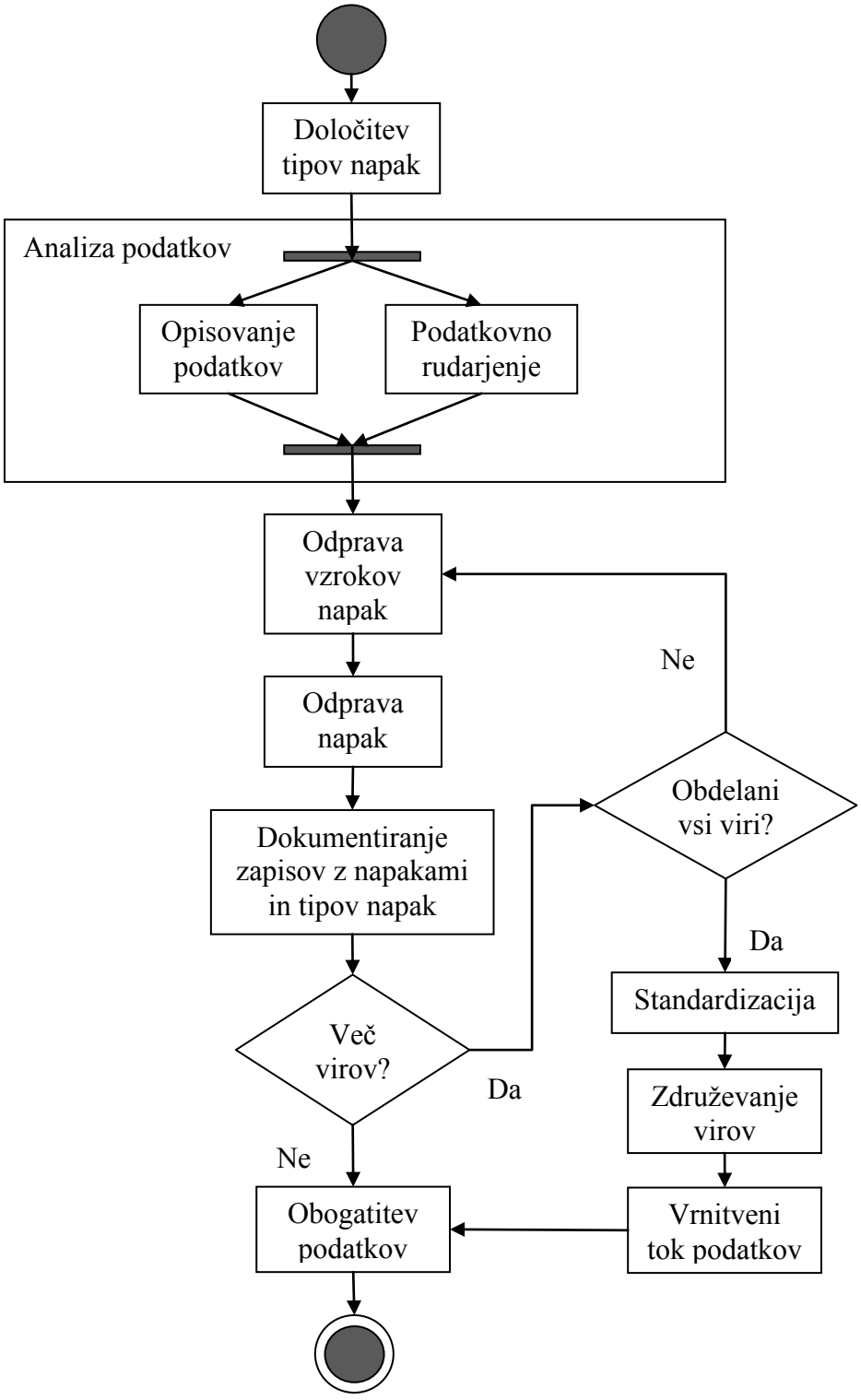
Ta faza vključuje izvedbo transformacijskih pravil (s pomočjo toka ETL ali pa z izvedbo na posameznih virih).

Vrnitveni tok čistih podatkov

Po zaključku čiščenja podatkov avtorja priporočata, da se očiščeni podatki prenesejo na vir in nadomestijo umazane podatke. Na ta način bodo tudi podedovani sistemi in aplikacije imele korist od čiščenja podatkov, hkrati pa ne bo potrebno ponovno čiščenje ob ponovnem izvozu podatkov.

6.2.4 Združene aktivnosti opisanih postopkov

V tej točki je predstavljen enoten postopek za čiščenje podatkov, ki sem ga sestavil iz zgoraj opisanih ogrodij [19, 34, 45]. Postopek na sliki 15 vsebuje vse poudarke omenjenih ogrodij, posamezne aktivnosti so opisane v opisih posameznih ogrodij v točki 6.2.3.



Slika 15: Diagram aktivnosti čiščenja podatkov

6.3 Metrike

Odsotnost metrik na tem področju bi za organizacijo predstavljalo oviro, saj brez njih ne moremo dokazati, kakšna je celovitost in posledično kakovost podatkov po uporabi akcij za njihovo izboljšanje [20]. V zvezi s kakovostjo podatkov predlaga literatura metrike z več področij, kar se sklada z definicijo in predpostavkami kakovosti podatkov. Ena skupina metrik se namreč lahko nanaša neposredno na podatke, druga skupina metrik pa mora obstajati za namen podatkovne varnosti.

Področje metrik je podrobneje opisal Loshin [29, 30], ki predlaga uporabo uravnoteženih kazalnikov kakovosti podatkov (ang. *balanced data quality scorecard*).

Sicer pa avtorji predlagajo uporabo metrik, ki se razlikujejo glede na vidik. Suer in Nolan [55] npr. z nivoja odločanja predlagata naslednji metriki:

- odstotek poročil, ki niso dostavljena pravočasno;
- odstotek poročil, ki vsebujejo nepravilnosti.

Z varnostnega vidika in vidika tveganj pa je Gelbstein [20] predlagal naslednje osnove metrik:

- popis pooblastil z dostopi in uporabnikov pooblastil;
- popis podatkov, ki se izvažajo in prenašajo v druge sisteme;
- število uporabnikov, ki so ohranili nekdanja pooblastila (z drugih sistemov itd.);
- število neaktivnih uporabniških računov;
- število aplikacij in sistemov, ki vsebuje vkodirane možnosti dostopa in stranska vrata;
- število primerov, ki so zahtevali dostop do produkcijskih podatkov in njihovo spremembo;
- število in delež nepooblaščenih dostopov ali sprememb produkcijskih podatkov;
- število incidentov, ki so povezani s podatki;
- število sistemov, ki niso povezani s sistemom identifikacije organizacije (IAM sistem);
- katalog nepravilnih ali neskladnih podatkov;
- odstotek podatkovnega modela organizacije, na osnovi katerega se meri celovitost;
- število metrik, ki so vključene v podatkovno bazo in aplikacije za zaznavo neskladnosti;
- število vgrajenih metrik za zaznavo nepooblaščenih dostopov do produkcijskih podatkov ali do operacijskih sistemov;
- število vgrajenih metrik za zaznavo sprememb, ki niso vključene v postopke za kontrolo sprememb;
- letna finančna izguba zaradi prevar, pri čemer je bil uporabljen informacijski sistem;
- število napadov na podatkovno celovitost na SCADA sistemih;
- število poročanj v tisku zaradi težav v celovitosti podatkov.

6.4 Obstoječe programske rešitve

Na trgu je na voljo mnogo programskih rešitev za čiščenje podatkov. Uporabimo lahko celovite rešitve, ki pokrivajo velik del aktivnosti čiščenja podatkov ali pa bolj usmerjene rešitve, namenjene reševanju točno določenega problema, odvisno od izvedbe postopka čiščenja podatkov in možnosti uporabe obstoječih programskih rešitev organizacije. Geiger [19] meni, da uporaba programskih rešitev za čiščenje podatkov bistveno pripomore k prihranku virov organizacije. Navaja primer organizacije, ki je po uvedbi programske opreme za izvedbo čiščenja podatkov v vsaki od sedmih enot letno prihranila tedensko delo 24-ih ljudi.

Na voljo so rešitve večjih podjetij, npr. Oracle [88], IBM [78], SAP [93]. Veliko pa je tudi manjših podjetij, ki ponujajo rešitve kot glavni ali pa enega izmed glavnih produktov.

Nekatere programske rešitve so namenjene specifičnim področjem. Na primer Chapman [9] je zbral pregled rešitev za področje biologije, druge pa so splošne, npr. *Data Match* [74].

Obstajajo pa tudi odprtokodne rešitve, npr. *Data Cleaner* [73], ter rešitve v obliki spletnih storitev [9].

Pregled nekaterih rešitev sta zbrala Rahm in Do [45]. Delita jih na več skupin:

- orodja za analizo podatkov, ki se delijo na:
  - orodja za opisovanje podatkov
  - orodja za podatkovno rudarjenje
- orodja za izvedbo čiščenja;
- specializirana orodja, ki se delijo glede na:
  - domeno oz. področje, npr. naslovi
  - vrsto transformacije, npr. čiščenje podvojenih zapisov
- orodja ETL.

Kot sta zapisala, število in raznolikost obstoječih rešitev nakazuje pomembnost in hkrati težavnost čiščenja podatkov. Ugotovila sta tudi, da je v večini primerov obstoječih rešitev njihova sposobnost vključitve v poljuben sistem omejena in da večina pokriva le del problematike ali del postopka čiščenja podatkov, zato je v teh primerih za želen rezultat čiščenja podatkov potrebna znatna količina ročnega dela ali programiranja.

Širok pregled rešitev in testov nudi spletna stran [77].

7. Predlog rešitve za čiščenje podatkov

7.1 Opis problemske domene in predloga rešitve

Pri delu v IT se pri uporabi relacijskih baz pogosto srečujemo s težavami glede kakovosti podatkov, strukturnimi težavami v bazi (npr. odsotnost ustreznih indeksov, ključev itd.), nekontrolirano rastjo posameznih tabel v smislu števila zapisov in podobnim. Poleg tega nepravilnosti v podatkih zaznavajo tudi poslovni uporabniki.

Da do takšnih težav ne pride, lahko v določeni meri poskrbita DBA in poslovni oz. sistemski analitik, ki imata dostop do relacijske baze. V obeh primerih je težava v tem, da bi se takšne kontrole izvajale le *ad hoc* in ne redno in kontrolirano. Poleg tega DBA najpogosteje razpolaga predvsem s tehničnim znanjem, poslovni analitik pa ima najpogosteje omejen dostop do sistemskih podatkov baze. Dodatna težava je v celostni odpravi odkritih težav. Pogosto se neka težava sicer zazna in odpravi napake v podatkih, vendar pa se kmalu spet pojavi, ker se ne odpravijo vsi viri težav.

Zaradi pogoste in ponavljajoče prisotnosti zgoraj opisanih težav pri svojem delu sem dobil zamisel o izgradnji lastne programske rešitve za samodejno zaznavo napak v podatkih v relacijski bazi (tako s strukturnega kot podatkovnega vidika) in obveščanje uporabnika.

Osnovni namen rešitve je ta, da uporabnika v najkrajšem možnem času opozori na prisotno nepravilnost. Od uporabnika je nato odvisno nadaljnje ukrepanje. Kaj predstavlja nepravilnost, določi uporabnik z vnosom v zbirko nepravilnosti. Možno je spremljati poljubno vsebino relacijske baze, ki je dostopna preko SQL jezika, npr.:

- skladnosti podatkov v različnih tabelah,
- vsebina posameznih tabel,
- različna štetja na osnovi poljubnih pravil,
- štetje zapisov v tabelah,
- strukturni podatki tabel in baze
- itd.

Razvil sem delujoči prototip [91] osrednjega dela ogrodja, ki je predstavljen v točki 7.4. Osnovno vodilo je bilo možnost splošne uporabe v katerem koli sistemu, torej izvedba z gradniki, ki jih je možno uporabiti praktično v katerem koli okolju, ki uporablja relacijsko podatkovno bazo, kjer so podatki dostopni preko poizvedovalnega jezika SQL. Jedro rešitve prototipa je periodično izvajanje poljubnih SQL poizvedb in obveščanje uporabnika, vendar le v primerih, v katerih uporabnik to želi – glede na rezultat poizvedb.

Prototip je predstavljen v točki 7.4. V točki 7.5 sem nato predlagal razširitev prototipa za možnost umestitve v informacijsko arhitekturo organizacije, kjer sem zaposlen. Opis organizacije je naveden v točki 7.2. Omenjena razširitev prototipa je zasnovana tako, da upošteva smernice, ki jih predlaga pregledana literatura [9, 20, 45], in so podrobneje navedene v točki 7.6. Zgradbo in možno umestitev v informacijski sistem organizacije sem prikazal s pogledi arhitekturnega jezika Archimate [70] s pomočjo orodja Archi [69]. Orodje je odprtokodno in prosto dostopno za modeliranje Archimate modelov. Razvijalo se je med letoma 2010 in 2012 kot del projekta v visokem šolstvu v Veliki Britaniji, na univerzi Bolton University. Od leta 2013 dalje pa se Archi razvija in vzdržuje na prostovoljni osnovi s strani avtorja, Phila Beauvoirja. Pri delu sem uporabljal navodila ter napotke za modeliranje z upoštevanjem najboljših praks [71].

Rešitev torej predstavlja ogrodje, ki uporabnikom nudi odkrivanje napak v podatkih. Vrste napak definirajo uporabniki sami preko vnosov poljubnih poizvedb. Ciljni uporabniki rešitve so:

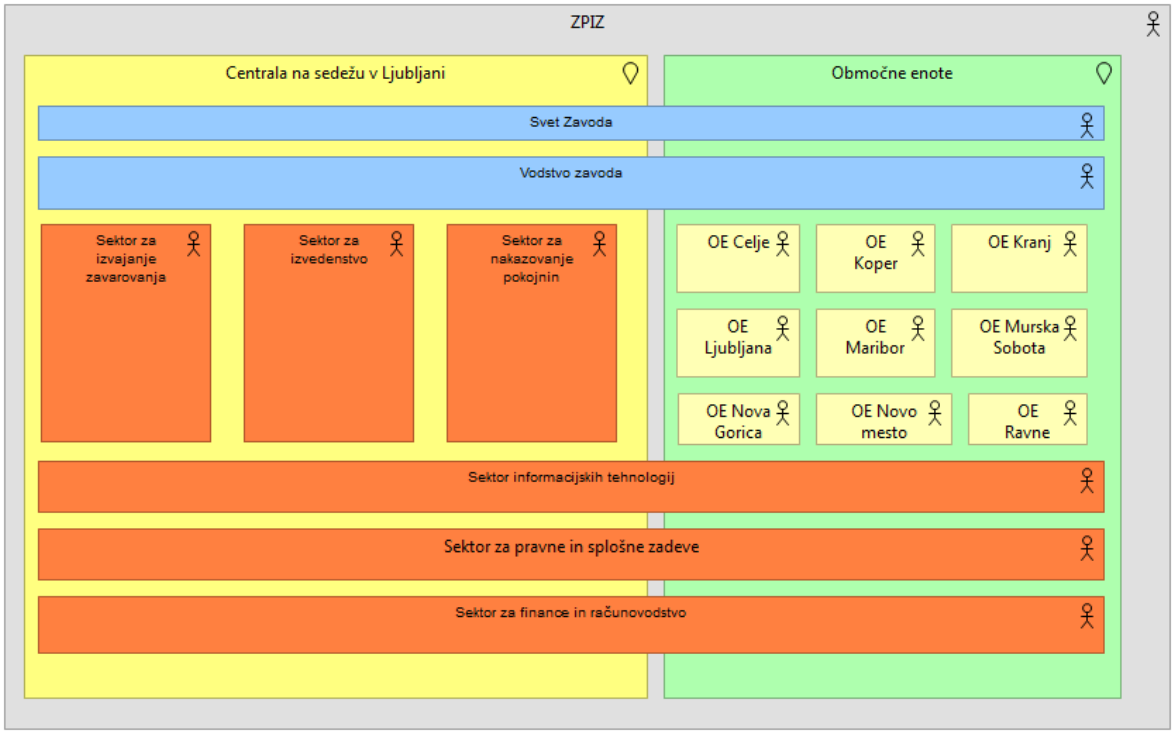
- poslovni skrbniki podatkov (tehnično naprednejši poslovni uporabniki, ki razpolagajo z znanji, navedenimi v točki 3.4.2),
- podatkovni skrbniki podatkov,
- sistemski analitiki oz. administratorji podatkovne baze.

V splošnem so to uporabniki, ki razpolagajo z znanjem poizvedovalnega jezika SQL.

Kot bom prikazal v nadaljevanju, predstavlja ogrodje tisti del upravljanja s kakovostjo podatkov DQM, ki se nanaša na zaznavanje napak v podatkih in posledično vzdrževanje pravilnosti podatkov, lahko pa se ga uporabi tudi za opisovanje (ang. *data profiling*), obravnavo napak oz. čiščenje podatkov ter izvajanje metrik. Odgovarja torej na tri problemska področja reaktivnega pristopa DQM oz. čiščenja podatkov ter na eno področje aktivnega pristopa DQM s slike 6.

7.2 Opis organizacije

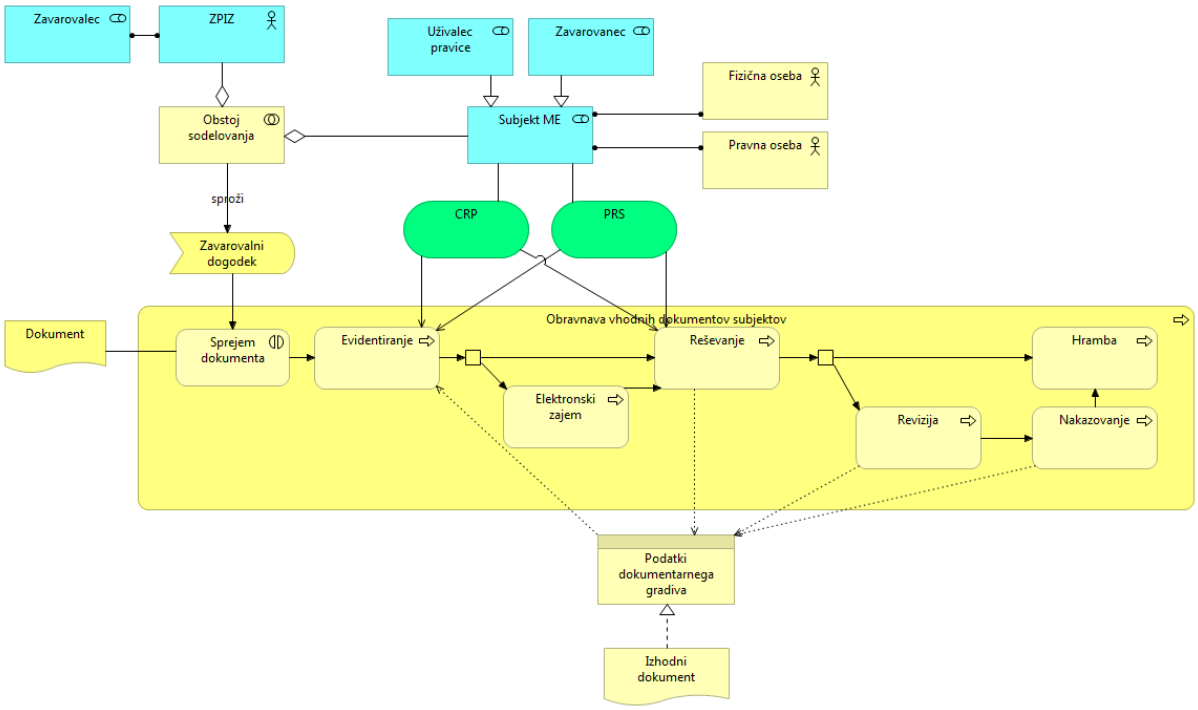
Zavod za pokojninsko in invalidsko zavarovanje (v nadaljevanju ZPIZ) je nosilec in izvajalec sistema pokojninskega in invalidskega zavarovanja v Sloveniji [97]. Model na sliki 16 predstavlja organigram ali organizacijsko strukturo Zavoda (t. i. makroorganizacijo).



Slika 16: Organizacijska shema zavoda

Spodnja slika 17 predstavlja osnovni poslovni proces temeljne dejavnosti zavoda. To je obravnava vhodnih dokumentov subjektov zavoda oz. reševanje zahtevkov strank. Pri tem kot subjekte obravnavamo tako pravne kot fizične osebe. Dokumenti pa so tako vloge za pridobitev pravic iz zavarovanja kot tudi dokumenti, ki jih zavezanci vlagajo po uradni dolžnosti (npr. prijave in odjave iz zavarovanja).

Do nepravilnosti v podatkih lahko pride v katerem koli od navedenih podprocesov. Pri tem lahko gre za napačno evidentiranje, napake v programski opremi, napačne vhodne podatke, pridobljene od zunanjih oseb, napake v podatkih iz podedovanih sistemov itd.



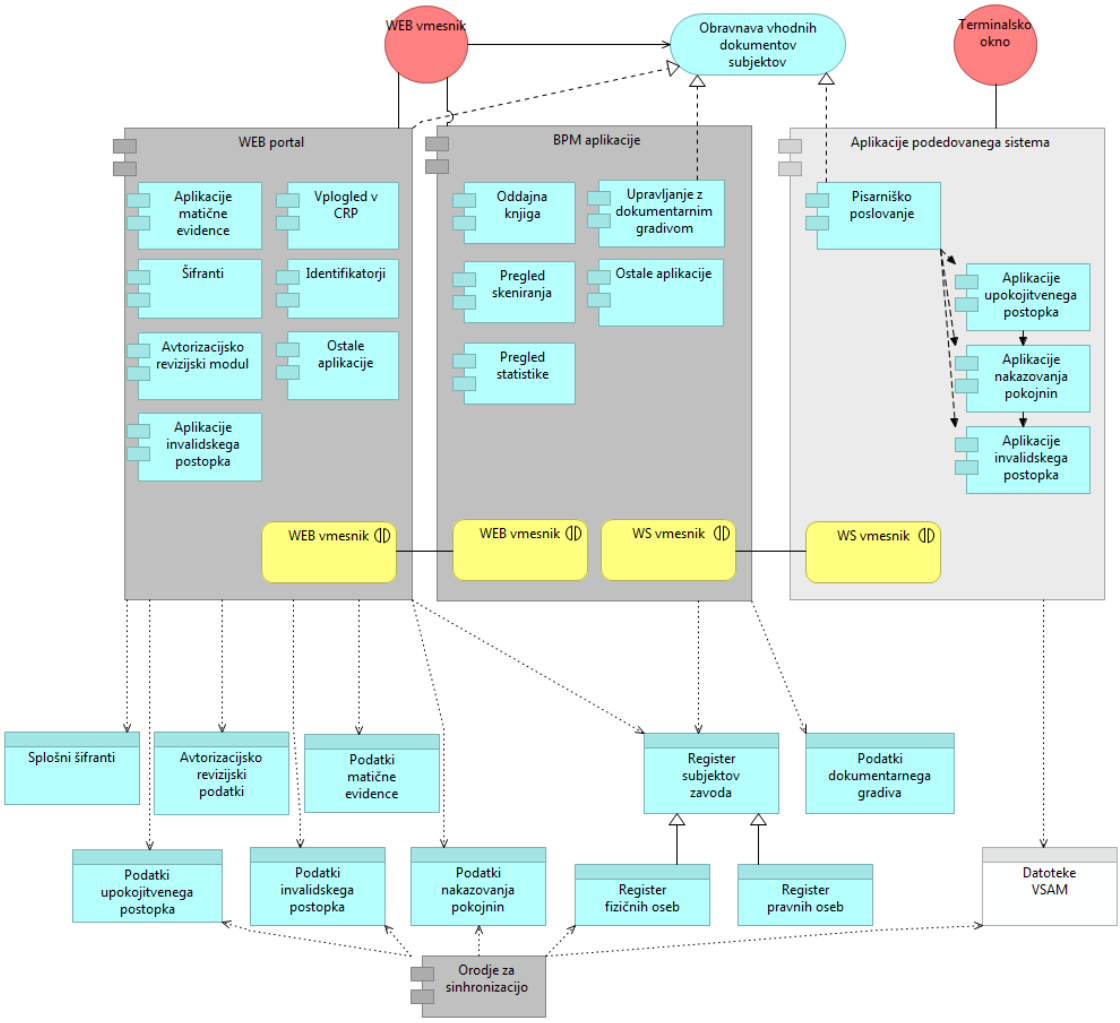
Slika 17: Osnovni poslovni proces temeljne dejavnosti zavoda

Spodnja slika 18 prikazuje povezanost aplikacijske storitve osnovnega poslovnega procesa zavoda z aplikacijskimi komponentami ter podatkovne zbirke, na katerih temelji njihovo delovanje. Modro obarvane zbirke so del relacijske baze, bela obarvana zbirka pa predstavlja datoteke podedovanega sistema. Iz slike je razvidna odvisnost aplikacijske storitve od podedovanega sistema. Analiza različnih napak v zavodu je pokazala vir tako v podedovanem sistemu kot tudi v novejšem sistemu.

Podedovani sistem je vir napak iz več vzrokov:

- neobstoj ustreznih kontrol v aplikacijah podedovanega sistema, kar je v preteklosti povzročilo nastanek različnih nepravilnosti v podatkih, od nepravilnih vrednosti posameznih podatkov do neskladnosti med redundantnimi podatki;
- tehnološke razlike med obema sistemoma, kar je ob selitvi podatkov v relacijsko bazo vir potencialnih težav;
- v preteklosti so različne organizacijske enote (glej sliko 16) imele ločene podatkovne zbirke. To pomeni, da so se istovrstni podatki vodili na več mestih, kar je vodilo v neskladnosti med njimi.

Informacijski sistem se sicer razvija in prenavlja, s čimer se uporaba podedovanega sistema zmanjšuje. V novem sistemu so z uporabo relacijske baze in enotnega podatkovnega modela določeni viri napak odpravljeni, klub temu pa napake nastajajo tudi v novem sistemu zaradi različnih vzrokov, opisanih v točki 4.



Slika 18: Model aplikacijskega nivoja

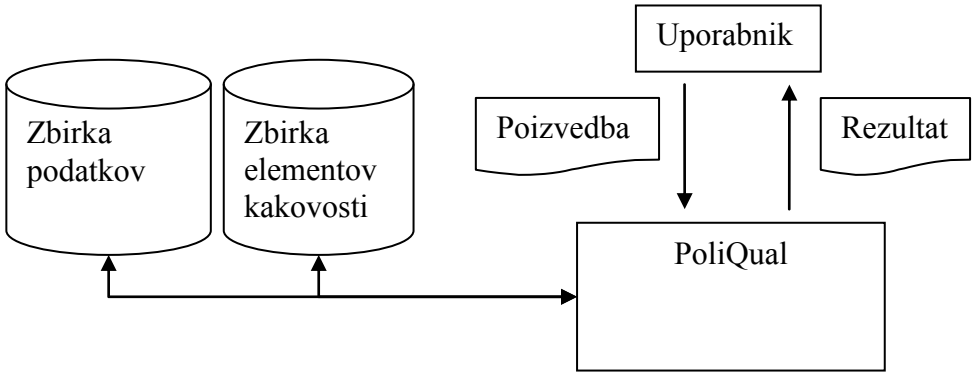
7.3 Obstoječe programske rešitve

V literaturi sem poskušal najti podoben primer pristopa k zaznavanju nepravilnosti v podatkovni bazi, opisovanju in čiščenju podatkov. Našel sem delo Cappiella, Francalancija in Pernicija[8], ki sicer ugotavljajo, da literatura ne vsebuje veliko predlogov za avtomatizacijo postopkov za zagotavljanje kakovosti podatkov ter da je njihovo delo prvo prizadevanje k zagotovitvi celostnega nabora orodij za sistematično upravljanje kakovosti podatkov, kjer je prostor še za številne nadaljnje raziskave. V osnovi uporabljajo podoben pristop k problemu in podoben namen ogrodja, razlikuje pa se v arhitekturnih elementih in v sami izvedbi. Razlikuje se na primer v naslednjem:

- način preverjanja ustreznosti zapisov v zbirki podatkov – omenjeno delo uporablja kazalnike kakovosti, shranjene v podatkovni zbirki [59], moja rešitev pa prepušča določitev celotnega pravila uporabniku;
- poizvedovalni jezik – omenjeno delo uporablja XML-QL poizvedovalni jezik [14], moja rešitev predvideva uporabo SQL;
- podatkovna zbirka – omenjeno delo v izhodišču predvideva možnost uporabe relacijske baze, v jedru analize pa uporablja model D^2Q [52], moja rešitev pa uporablja relacijsko bazo;
- možnost uporabe bolj naprednih poizvedb (npr. povezovanje več tabel) – omenjeno delo navaja, da ni predvideno za takšno uporabo, moja rešitev pri tem nima omejitev oz. je omejena le z zunanjimi omejitvami (morebitne nastavitve in omejitve s strani DBA);
- posredovanje administratorja kakovosti v primerih popravkov podatkov – moja rešitev takšnega posredovanja ne potrebuje;
- predstavitev rezultatov in predvideno proženje orodja.

V omenjenem delu [8] so avtorji predstavili ogrodje PoliQual za spremljanje podatkov. Vključuje ocenjevanje kakovosti podatkov in predlog za popravek podatkov (ta del zahteva posredovanje administratorja). Metoda kot vhodni podatek potrebuje stopnjo kakovosti podatkov za vsako množico podatkov oz. vsako posamezno poizvedbo, določeno s strani uporabnika. Stopnjo kakovosti določajo vrednosti treh elementov kakovosti: natančnost oz. pravilnost, popolnost in pravočasnost. Gre torej za uporabo kazalnikov kakovosti [59].

Ti trije kazalniki kakovosti so zapisani v posebni zbirki kakovosti za vsak osnovni zapis v zbirki podatkov, kot prikazuje slika 19. Njihov model je zasnovan na osnovi modela Data and Data Quality (D^2Q) [52] in uporablja poizvedovalni jezik XML-QL [14].



Slika 19: Sistem PoliQual

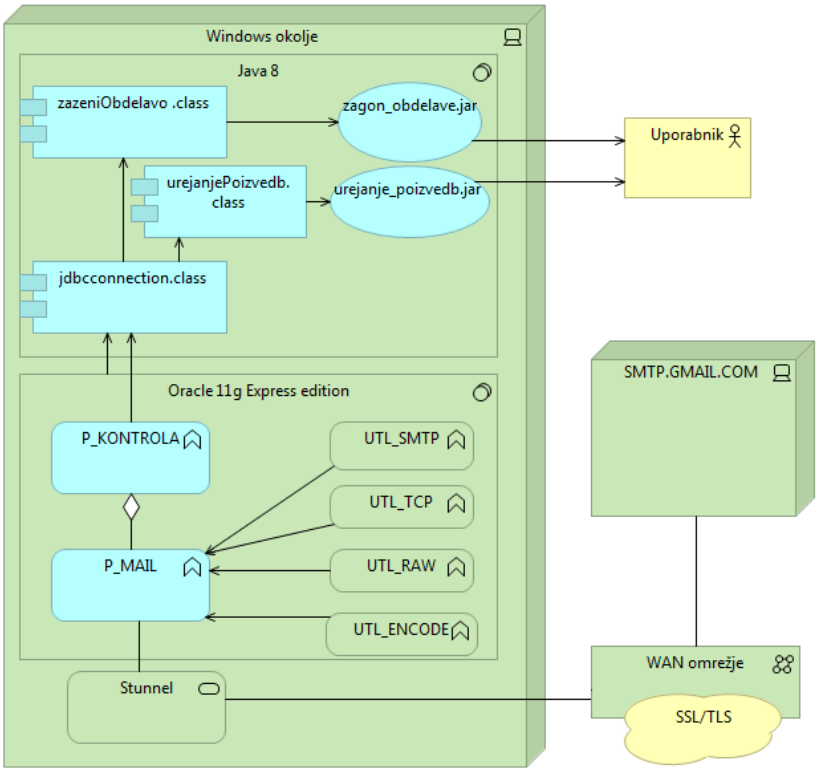
7.4 Opis prototipne rešitve

V tej točki je predstavljeno delovanje prototipne rešitve ter njene posamezne komponente.

7.4.1 Arhitekturni model

Na sliki 20 je prikazan arhitekturni model, izdelan v arhitekturnem jeziku Archimate. Delovanje rešitve je naslednje: uporabnik preko uporabniškega vmesnika *urejanje_poizvedb.jar* v podatkovno bazo v zbirko kontrolnih poizvedb vnese poljubne poizvedbe, za katere želi, da se periodično izvajajo in preverjajo želeno vsebino relacijske baze. Nato vnese še elektronski naslov, na katerega želi prejeti obvestilo v primeru, da dejanski rezultat poizvedbe odstopa od pričakovanega rezultata. Uporabniški vmesnik je pri tem delu le v pomoč, vnos lahko uporabniki naredijo tudi neposredno v tabeli relacijske baze, prikazani na sliki 21.

Uporabniški vmesnik *zagon_obdelave.jar* služi za zagon obdelave kontrolnih poizvedb (v produkcijskem okolju bi se v ta namen uporabilo zagon obdelav po urniku). Procedura *P_KONTROLA* poišče vse kontrolne poizvedbe, ki se morajo izvesti, jih izvede ter glede na rezultat pošlje obvestilo uporabnikom preko procedure *P_MAIL*. Posamezni gradniki rešitve so podrobneje opisani v nadaljevanju.



Slika 20: Arhitekturni diagram prototipa

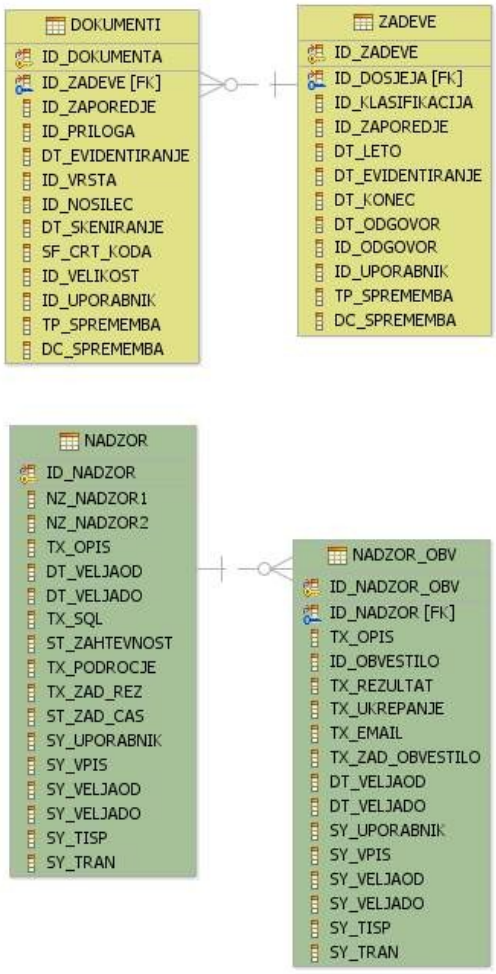
7.4.2 Opis posameznih komponent

Oracle 11g Express Edition

Oracle 11g Express Edition (v nadaljevanju *Oracle XE*) [88] je relacijska baza, ki sem jo uporabil za izdelavo prototipa. Spodnji diagram ER prikazuje podatkovni model za podporo prototipa. Z zeleno sta označeni tabeli, ki ju orodje uporablja za izvedbo iskanja napak:

- tabela NADZOR predstavlja zbirko kontrolnih poizvedb,
- tabela NADZOR_OBV predstavlja zbirko naročil na kontrolne poizvedbe.

Ostali dve tabeli pa sta primer poslovnega dela podatkovnega modela, nad katerim lahko izvajamo testne poizvedbe za iskanje napak. Poizvedbe lahko izvajamo nad katerokoli tabelo v relacijski bazi. Za namen prikaza delovanja prototipa sta v tej točki za poslovni del podatkovnega modela uporabljeni dve izmed tabel, ki podpirata delovanje Sistema za upravljanje z zadevami ZPIZ, ki je obravnavan v točki 7.5. Imena teh dveh obstoječih objektov so zaradi varnosti spremenjena in ne odražajo dejanskega stanja.



Slika 21: Podatkovni model za podporo prototipa

Stunnel 5.22

Stunnel [94] je orodje, ki služi kot vmesnik med podatkovno bazo *Oracle XE* in strežnikom za pošiljanje elektronske pošte SMTP. Uporabljen je zato, ker večina strežnikov za pošiljanje elektronske pošte SMTP zahteva varno povezavo TLS/SSL. V prototipu sem uporabil strežnik za pošiljanje pošte SMTP.GMAIL.COM, ki je dostopen na vratih 465.

Uporabil sem takšno nastavitev *Stunnela*, ki sprejema pakete na lokalnem IP 127.0.0.1 na vratih 1925 in jih po varni povezavi preusmeri na SMTP.GMAIL.COM. Nastavitev se vnese v sistemsko datoteko *Stunnel.conf*:

[gmail-smtp]
client = yes
accept = 127.0.0.1:1925
connect = smtp.gmail.com:465

Preprost test delovanja povezave je naslednji: v terminalsko okno vnesemo ukaz:

telnet localhost 1925 ,

s čimer bo stekla komunikacija preko *Stunnela.* Če se nam prikaže pozdravno besedilo s strežnika SMTP.GMAIL.COM, je povezava delujoča. V nasprotnem primeru, ko povezave ni mogoče vzpostaviti in če nimamo možnosti spreminjanja mrežnih nastavitev, lahko z aplikacijo prav tako delamo. Le da bomo v tem primeru sporočilo dobili na uporabniški vmesnik in ne na elektronsko pošto, kot je opisano v nadaljevanju.

Druga možnost vzpostavitve varne povezave med podatkovno bazo *Oracle 11g Express Edition* in strežnikom za pošiljanje pošte SMTP je uporaba ustrezno nastavljene funkcionalnosti *Oracle XE Wallet*. Oracle preko omenjene funkcionalnosti namreč podpira varno povezavo TLS/SSL z zunanjimi strežniki. Za nastavitev potrebujemo orodje *Oracle Wallet Manager*, ki pa ni prosto dostopno.

Java 8

Za izdelavo in poganjanje uporabniškega vmesnika sem uporabil Javo verzije 8 [86].

Strežnik SMTP.GMAIL.COM

Strežnik sem uporabil za pošiljanje elektronskih sporočil. Zahteva:

- varno TLS/SSL povezavo (v ta namen je bilo uporabljeno predhodno opisano orodje *Stunnel*);
- kodiranje podatkov (v ta namen sem uporabil *Oracle XE* paket *UTL_ENCODE*);
- overjanje uporabnika (v ta namen sem izdelal GMAIL račun).

Komunikacija poteka preko vrat 465.

Razred Java jdbcconnection

Razred služi za komunikacijo med uporabniškimi vmesniki *Java* in lokalno podatkovno bazo *Oracle XE*.

Metode, ki jih vsebuje:

- *start_P_KONTROLA* (uporablja se za zagon obdelave – PL/SQL procedure *P_KONTROLA*);
- *preberiVrednosti* (uporablja se za izvedbo stavka SQL *Select*, prejetega v vhodnem parametru, kot odgovor pa vrne rezultat iz podatkovne baze – en zapis);
- *vnosPoizvedbe* (uporablja se za izvedbo SQL stavkov *Update* ali *Insert* ali *Delete* ter potrjevanje *Commit* ali razveljavitev *Rollback*);
- *napolniPrikaznoPolje* (uporablja se za izvedbo stavka SQL *Select*, prejetega v vhodnem parametru, kot odgovor pa vrne rezultat iz podatkovne baze – več zapisov).

Razred Java zazeniObdelavo

Razred je realiziran kot izvršljiva datoteka *zagon_obdelave.jar*. Uporabljen je za izdelavo grafičnega uporabniškega vmesnika, ki služi za zagon obdelave (procedure PL/SQL *P_KONTROLA*) s pomočjo metode *jdbcconnection.start_P_KONTROLA*.

Razred Java urejanjePoizvedb

Razred je realiziran kot izvršljiva datoteka *urejanje_poizvedb.jar*. Uporabljen je za izdelavo grafičnega uporabniškega vmesnika, ki služi za

- pregledovanje kontrolnih poizvedb,
- pregledovanje naročil na rezultate poizvedb,
- urejanje (vnos in odstranitev) kontrolnih poizvedb,
- urejanje (vnos in odstranitev) naročil na rezultate poizvedb.

7.4.3 Proceduri PL/SQL

7.4.3.1 Procedura P_KONTROLA

P_KONTROLA je procedura PL/SQL, ki poišče nabor aktualnih poizvedb (to pomeni, da ustrezajo datumskemu kriteriju in da zanje obstaja naročilo uporabnika) in jih izvede. Nato rezultate pošlje na vse elektronske naslove, ki so naročeni na posamezno poizvedbo. Kot rezultat vrne obvestilo o uspehu obdelave, število uspešno poslanih elektronskih sporočil, število vseh pripravljenih elektronskih sporočil in seznam neuspešno poslanih elektronskih sporočil.
Vhodni parametri za proceduro niso predvideni. Izhodni parametri pa so naslednji:

- obvestilo o uspehu obdelave,
- število uspešno poslanih elektronskih sporočil,
- število vseh pripravljenih elektronskih sporočil,
- seznam neuspešno poslanih elektronskih sporočil.

7.4.3.2 Procedura P_MAIL

Vhodni parametri procedure so naslednji:

- številka transakcije,
- število elektronskih sporočil.

Izhodni parametri so naslednji:

- obvestilo o uspehu obdelave,
- število uspešno poslanih elektronskih sporočil,
- seznam uspešno poslanih elektronskih sporočil.

P_MAIL je procedura PL/SQL, ki komunicira s strežnikom SMTP in glede na vhodni parameter številke transakcije poišče podatke in pošlje ustrezna sporočila. Znotraj te procedure se uporabijo paketi baze *Oracle XE*:

- *UTP_SMTP* za komunikacijo s strežnikom za pošiljanje elektronske pošte,
- *UTL_TCP* za oblikovanje besedila (CRLF) pri sestavljanju elektronskega sporočila,
- *UTL_RAW* za pretvorbo besedila v ustrezen format, ki ga je mogoče zakodirati,
- *UTL_ENCODE* za kodiranje besedila.

7.4.3.3 Vsebina procedur

Spodnja psevdokoda predstavlja jedro sistema za izvajanje kontrolnih poizvedb – vsebino obeh procedur na pregleden in jedrnat način.

1) Izdelava seznama poizvedb

Pripravimo seznam poizvedb, ki jih bomo pognali:

```
--SQL1
Select * from NADZOR a
Where a.SY_TISP != 'B'
And (a.SY_VELJADO is null or a.SY_VELJADO >= systimestamp)
And (a.DT_VELJADO is null or a.DT_VELJADO >= systimestamp)
And exists (    select 1 from NADZOR_OBV b
                Where b.SY_TISP != 'B'
                And (b.SY_VELJADO is null or b.SY_VELJADO >= systimestamp)
                And (b.DT_VELJADO is null or b.DT_VELJADO >= systimestamp)
                And b.ID_NADZOR = a.ID_NADZOR )
Order by a.TX_PODROCJE ;
```

Če je rezultat SQL1 nič zapisov, končaj obdelavo, sicer nadaljuj.

2) Izvajanje poizvedb

Pognali bomo toliko poizvedb, kolikor nam jih je *SQL1* vrnil, recimo temu *SQL1[x=1..n]*. Posledično bomo kasneje poslali določeno število elektronskih sporočil.

Za vsak zapis *SQL1[x]* izvedemo:

Preberemo *SQL1[x].TX_SQL* in izvedemo poizvedbo SQL, ki je zapisana v prebranem polju. Rezultat shranimo v spremenljivko *:rezultat*.

Rezultat poizvedbe zapišemo v tabelo *NADZOR* v obravnavano vrstico, v stolpec *TX_ZAD_REZ:*

```
Update NADZOR
Set SY_TISP = 'S' ,
SY_VPIS = systimestamp,
SY_UPORABNIK = user ,
SY_TRAN = (select max(SY_TRAN) + 1 from NADZOR),
TX_ZAD_REZ = :rezultat – (rezultat izvedenega SQL[x]
ST_ZAD_CAS = :čas zvajanja SQL v milisekundah
Where ID_NADZOR = SQL1[x].ID_NADZOR ;
```

Konec zanke.

3) Priprava vsebine elektronskih sporočil

Pripravimo vsebino elektronskih sporočil za vse predhodno pognane SQL.

--SQL2
Select b. from NADZOR a , NADZOR_OBV b*
Where a.SY_TISP != 'B'
And (a.SY_VELJADO is null or a.SY_VELJADO >= systimestamp)
And (a.DT_VELJADO is null or a.DT_VELJADO >= systimestamp)
And a.ID_NADZOR = b.ID_NADZOR
And a.SY_TRAN = :transakcija, ki se izvaja – shranjena predhodno v tabeli NADZOR;

Za vsak zapis *SQL2[x]* pripravimo besedilo elektronskega sporočila, v odvisnosti od pričakovanega rezultata:

Obvestilo pripravimo le v primeru, ko ga je naročnik želel, kar je odvisno od nastavitve v *NADZOR_OBV.ID_SPOROCILO* :

1 – obvestilo ne glede na rezultat,
2 – obvestilo, kadar je rezultat SQL različen od pričakovanega rezultata,
3 – obvestilo, kadar je rezultat SQL enak pričakovanemu rezultatu.

Če obravnavani zapis ustreza zgornjemu pogoju, sestavimo obvestilo, sicer gremo na naslednji zapis v zanki.

Obvestilo sestavimo na naslednji način:

ŠTEVILKA TRANSAKCIJE: :transakcija, ki se izvaja,
PODROČJE: področje, ki ga preverja SQL (iz tabele NADZOR)
UKREPANJE: podatek o pomembnosti ukrepanja (iz tabele NADZOR_OBV)
REZULTAT SQL: rezultat SQL (iz tabele NADZOR)
PRIČAKOVANI REZULTAT: rezultat, ki ga pričakuje posamezni naročnik obvestila (iz tabele NADZOR_OBV),
KRATKA_VSEBINA: kratek opis preverjanja (iz tabele NADZOR)

Pripravljeno obvestilo zapišemo v obravnavano vrstico *NADZOR_OBV*:

Update NADZOR_OBV
set SY_VPIS = systimestamp ,
SY_UPORABNIK = user ,
SY_TISP = 'S' ,
SY_TRAN = :transakcija, ki se izvaja ,
TX_ZAD_OBVESTILO = :sestavljeno obvestilo
Where ID_NADZOR_OBV = SQL[x].ID_NADZOR_OBV ;

Konec zanke

4) Pošiljanje obvestil

Ko zaključimo z izvajanjem poizvedb, pošljemo vsa nastala obvestila (vsebino iz *TX_ZAD_OBVESTILO*) na elektronske naslove. To naredimo tako, da za vse vrstice, ki so bile spremenjene v prejšnji točki, pošljemo vsebino *TX_ZAD_OBVESTILO* na *TX_EMAIL*.

5) Zaključimo obdelavo

Zaključimo obdelavo in v izhodne parametre vrnemo vrednosti glede na spodnjo preglednico 3.

Polje	Tip	Opis
odgovor	*string*	*ERR* – v obdelavi je prišlo do tehnične napake. *OK* – v obdelavi ni prišlo do tehnične napake.
steviloPoslanih	*int*	Število uspešno poslanih elektronskih sporočil.
steviloVsehEmailov	*int*	Število vseh elektronskih sporočil za pošiljanje.
seznamNeuspesnoPoslanih	*string*	Seznam identifikatorjev, za katere ni bilo mogoče poslati elektronskega sporočila.

Preglednica 3: Izhodni parametri obdelave

7.4.4 Uporabniška vmesnika

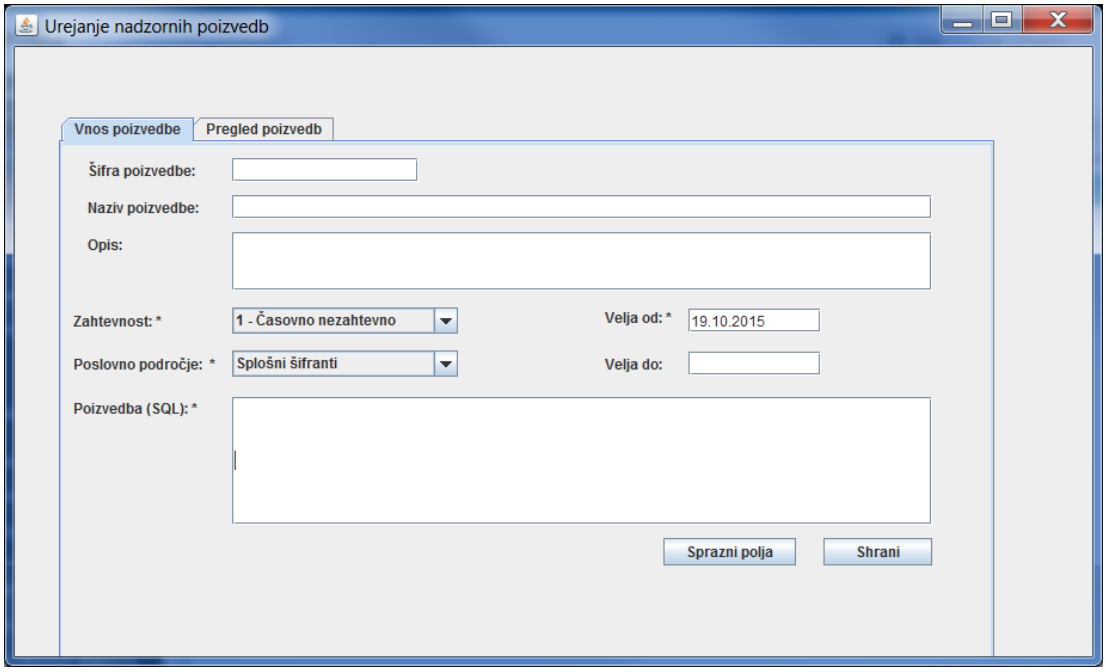
Prototip zajema dva uporabniška vmesnika, izdelana v orodju *Eclipse Luna* v jeziku *Java*:

- *urejanje_poizvedb.jar* je uporabniški vmesnik za urejanje poizvedb in naročil nanje, znotraj tega vmesnika se nahaja več oken;
- *zagon_obdelave.jar* je uporabniški vmesnik za proženje kontrolnih poizvedb (proženje naj bi se v produkcijski uporabi izvajalo po določenem urniku, v prototipu pa je ta del simuliran s pomočjo preprostega uporabniškega vmesnika, kjer se obdelava proži na zahtevo).

7.4.4.1 Vmesnik za urejanje poizvedb in naročil

7.4.4.1.1 Zavihek Vnos poizvedbe

Uporabniški vmesnik s slike 22 se uporabi za vnos nove kontrolne poizvedbe. Polja, označena z zvezdico (*) so obvezna. Pomen posameznih polj in gumbov je opisan spodaj.



Slika 22. Uporabniški vmesnik za vnos nove kontrolne poizvedbe

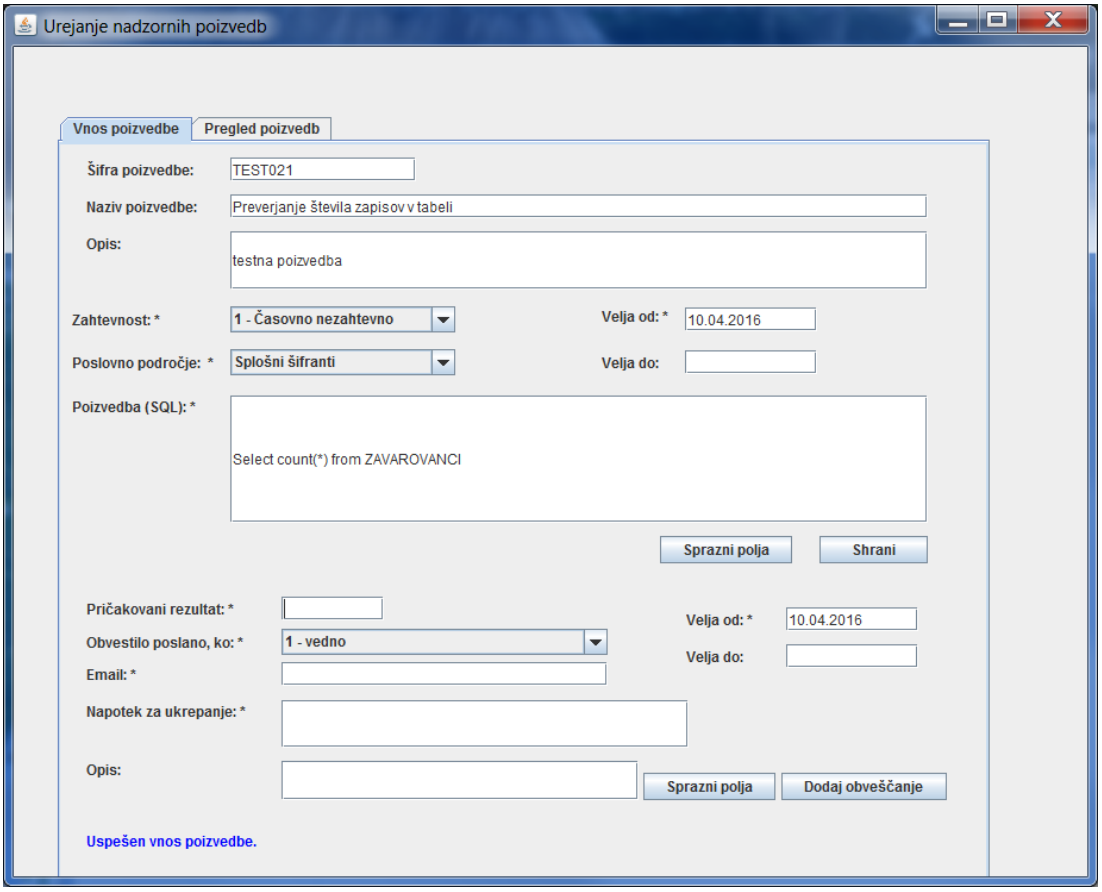
Šifra poizvedbe: poljubna šifra poizvedbe, dolžine največ 20 znakov.
Naziv poizvedbe: poljuben naziv poizvedbe, dolžine največ 250 znakov.
Opis: poljuben opis poizvedbe, dolžine največ 500 znakov.
Zahtevnost: izberemo podatek o zahtevnosti poizvedbe. Podatek je uporaben za možnost morebitnega izločanja ali razvrščanja poizvedb po časovni zahtevnosti, če bi se pojavilo preveč poizvedb ali bi nastala težava s sistemskimi viri.

Poslovno področje: izberemo poslovno področje, na katero se poizvedba nanaša. To je uporabno za pregled poizvedb po poslovnih področjih.
Poizvedba: vnesemo poizvedbo SQL, dolžine največ 2000 znakov, brez znaka ; na koncu. V prototipu je predvideno, da poizvedbe vračajo število. Primer ustreznega vnosa:

select count(*) from ZAVAROVANCI

Velja od: datum, od katerega se bo poizvedba uporabljala ob izvajanju obdelave, v formatu *dd.mm.llll*. Privzet je sistemski datum, ki ga lahko poljubno spremenimo.
Velja do: datum, do katerega se bo poizvedba uporabljala ob izvajanju obdelave, v formatu *dd.mm.llll*.
Gumb **Sprazni polja** izprazni vsa polja za vnos nove poizvedbe.
Gumb **Shrani** vnesene vrednosti shrani v tabelo poizvedb NADZOR. Če je zapis uspešno shranjen, se nam ponudijo polja za dodajanje naročila na poizvedbo, opisana spodaj.

Polja za vnos naročila na rezultate poizvedbe predstavlja spodnja skupina polj na sliki 23. Polja, označena z zvezdico (*), so obvezna. Pomen posameznih polj in gumbov je opisan spodaj.



Slika 23: Uporabniški vmesnik za vnos naročila na rezultate kontrolne poizvedbe

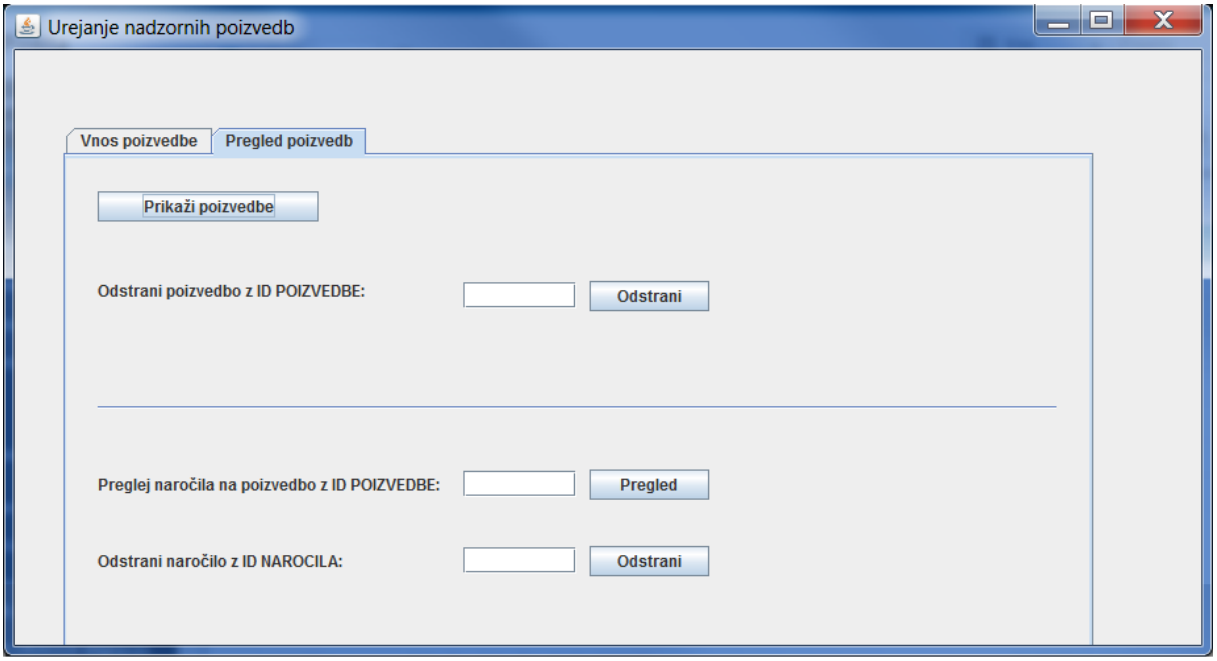
Pričakovani rezultat: številčna vrednost – rezultat, ki ga pričakujemo za poizvedbo z zgornjega dela ekrana (pravilen rezultat, ki ne odraža napak).
Obvestilo poslano, ko: izberemo, v katerem primeru želimo, da nam sistem pošlje obvestilo; na voljo imamo:

- vedno,
- ko se rezultat razlikuje od pričakovanega,
- ko je rezultat enak pričakovanemu,
- nikoli.

Email: elektronski naslov, na katerega bo poslano obvestilo.
Napotek za ukrepanje: besedilo, ki ga prejme prejemnik elektronskega sporočila.
Opis: poljuben opis, ki ga prejme prejemnik elektronskega sporočila.
Velja od: datum, od katerega se bo naročilo na poizvedbo upoštevalo, v formatu *dd.mm.llll.* Privzet je sistemski datum, ki ga lahko poljubno spremenimo.
Velja do: datum, do katerega se bo naročilo na poizvedbo upoštevalo, v formatu *dd.mm.llll*.
Gumb **Sprazni polja** izprazni vsa polja za vnos nove poizvedbe.
Gumb **Dodaj obveščanje** shrani vnesene vrednosti v tabelo poizvedb NADZOR_OBV.

7.4.4.1.2 Zavihek Pregled poizvedb

Uporabniški vmesnik s slike 24 se uporabi za prikaz seznama kontrolnih poizvedb, seznama naročil nanje ter za odstranitev posamezne poizvedbe in naročil. Pomen posameznih polj in gumbov je opisan spodaj.



Slika 24: Uporabniški vmesnik za pregled in brisanje kontrolnih poizvedb in naročil

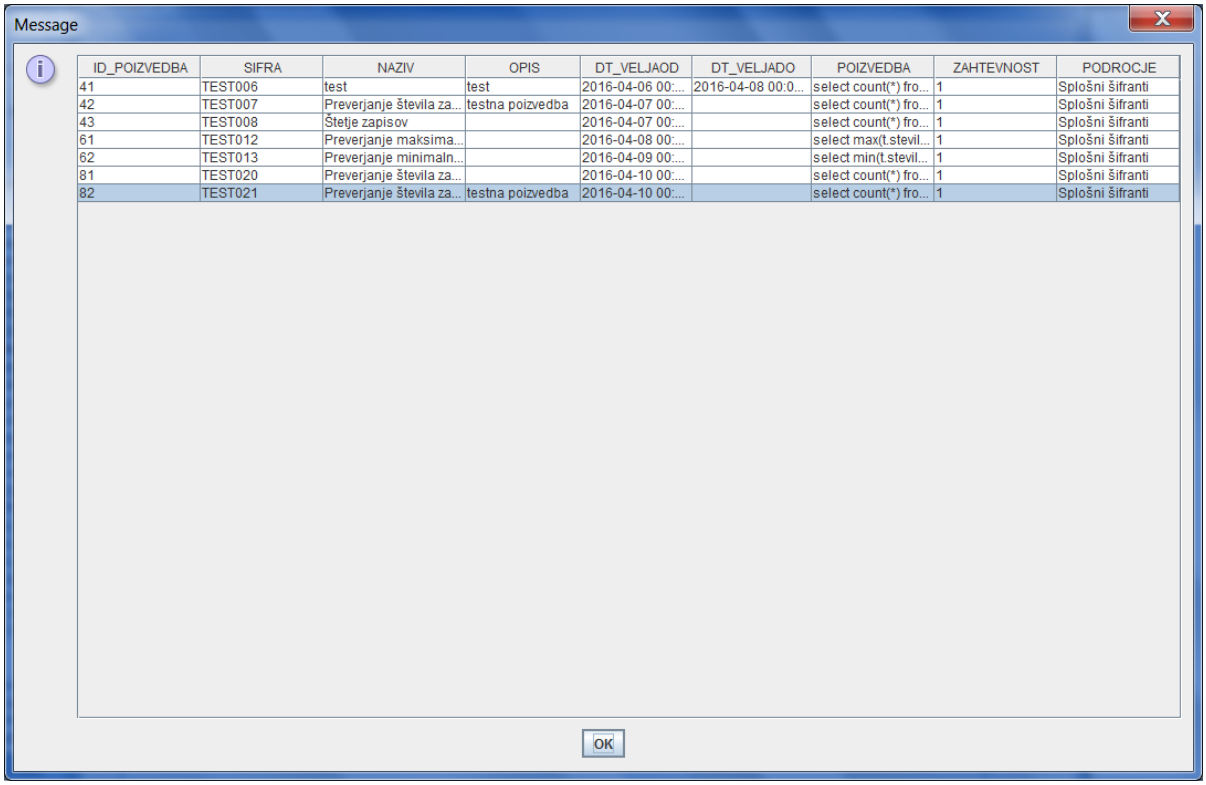
Zgornji vmesnik se uporabi za pregled in odstranjevanje obstoječih poizvedb.

Gumb **Prikaži poizvedbe** nam v novem oknu, prikazanem na sliki 25, prikaže vse kontrolne poizvedbe iz tabele NADZOR, razen logično brisanih, ki smo jih predhodno odstranili z gumbom **Odstrani**.

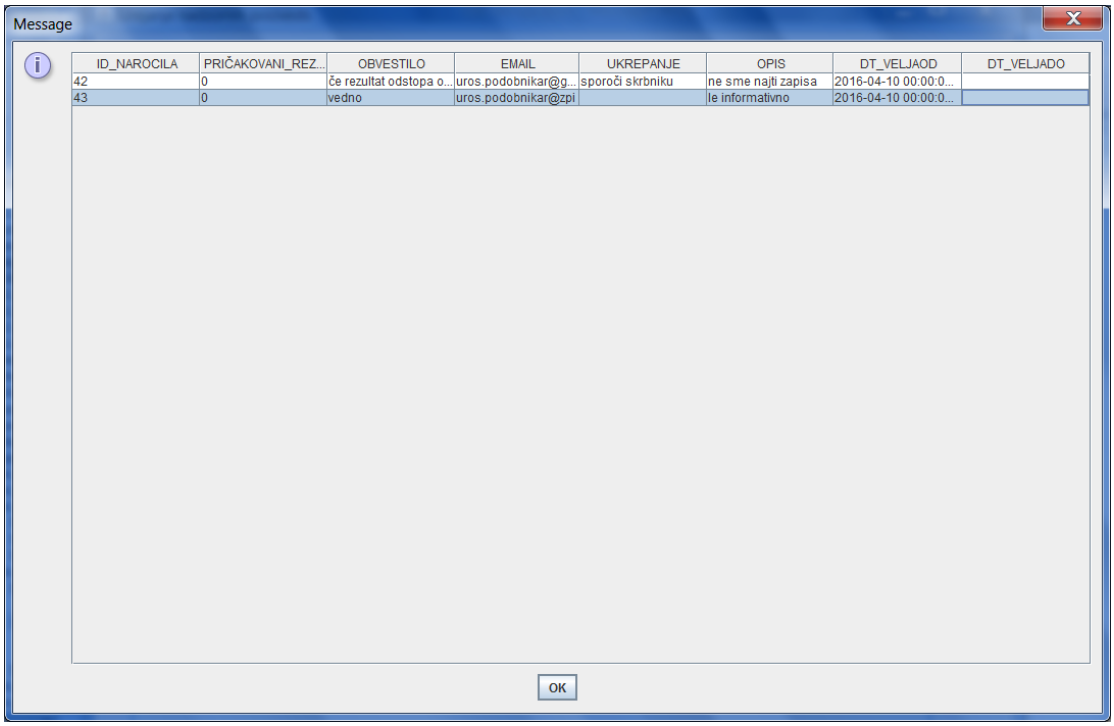
Prvi gumb **Odstrani** logično briše kontrolno poizvedbo, katere identifikator je vnesen v polju pred gumbom (identifikator smo lahko prebrali iz predhodnega pregleda poizvedb).

Naročila na določeno poizvedbo pregledamo tako, da v polje pred gumbom **Pregled** vnesemo ustrezni identifikator poizvedbe, nato pa uporabimo gumb **Pregled**. Prikaže se nam novo okno, prikazano na sliki 26.

Drugi gumb **Odstrani** logično briše naročilo, katerega identifikator je vnesen v polju pred gumbom.



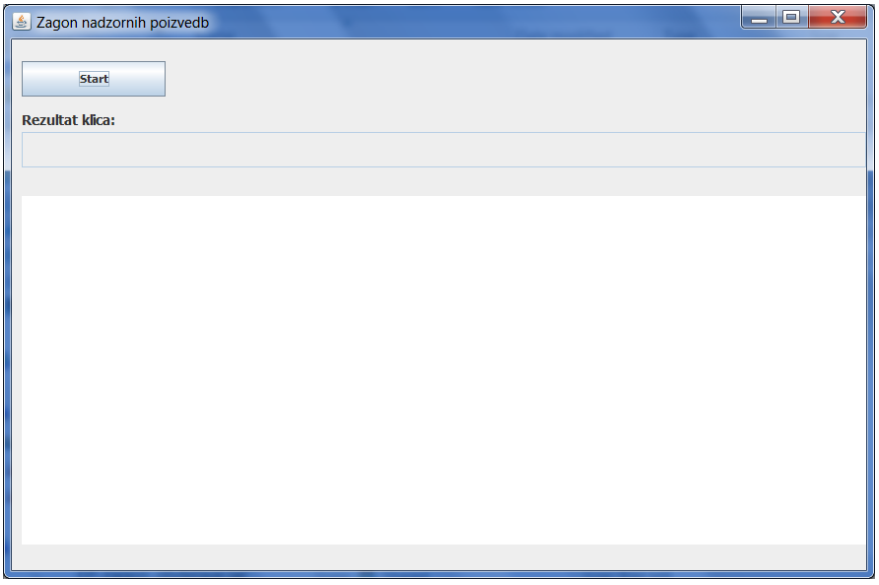
Slika 25: Uporabniški vmesnik za pregled obstoječih kontrolnih poizvedb



Slika 26: Uporabniški vmesnik za pregled naročil na rezultate kontrolne poizvedbe

7.4.4.2 Vmesnik za zagon poizvedb

Ta vmesnik se uporabi za zagon obdelave za izvedbo kontrolnih poizvedb – kot simulacija izvajanja po urniku v produkcijskem okolju. V poljih na vmesniku se izpiše rezultat obdelave. V primeru, da prototip nima povezave s spletom oz. s strežnikom za pošiljanje elektronske pošte, se vsebina elektronskih sporočil izpiše v tem oknu.



Slika 27: Uporabniški vmesnik za zagon obdelave

7.4.5 Uporabljena razvojna orodja

Eclipse Luna

Eclipse Luna [75] je orodje, uporabljeno za izdelavo grafičnih vmesnikov in kode java. Za dostop do podatkovne baze iz kode java je potrebno v programski projekt uvoziti ustrezni gonilnik za povezavo s podatkovno bazo, za ta namen sem uporabil gonilnik *ojdbc6.jar*. Naslednja koda nam nato uvozi paket za dostop do podatkovne baze neposredno iz programske kode:

```
import java.sql.*
```

Oracle SQL Developer

Oracle SQL Developer [88] je orodje za delo z bazo podatkov. Uporabil sem ga za nastavitve podatkovne baze, dodelitev pooblastil, izdelavo tabel, procedur PL/SQL, pripravo testnih tabel in podatkov.

Za delo sem izdelal in uporabljal uporabnika oz. shemo *naloga*. Posebne nastavitve, ki jih je bilo treba uporabiti:

- Uporabniku *naloga* dodeliti pooblastila za sistemske pakete (za namen prototipa sem uporabil širši obseg pooblastil, kot bi bilo potrebno v produkcijskem okolju):

```
grant all on UTL_HTTP to naloga ;
grant all on UTL_TCP to naloga ;
grant execute on UTL_ENCODE to naloga ;
grant execute on UTL_SMTP to naloga ;
```

- Vzpostavitev seznama dostopov ACL – dovoljenja uporabniku *naloga* za dostop do mreže oz. spleta:

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.CREATE_ACL (
    acl => 'gmail.xml',
    description => 'Permissions for smtp gate',
    principal => 'NALOGA',
    is_grant => TRUE,
    privilege => 'connect'
    );
```

```
COMMIT;
END;

BEGIN
        DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL (
        acl => 'gmail.xml',
        host => 'localhost',
                --localhost zato, ker bo tu poslušal Stunnel za posredovanje
                -- na smtp.gmail.com (glej opis pri Stunnel)
        lower_port => null,
        upper_port => null );
COMMIT;
END;
/
```

7.4.6 Prikaz delovanja na primeru

Spodnji primer ponazarja delovanje prototipa na možnem realnem problemu v poslovnem okolju. Za primer vzemimo podatkovni model, kot je bil prikazan na sliki 21. Uporabnik si na primer želi, da je obveščen v primeru, ko se v podatkovni bazi pojavi zapis o zadevi, ki ji ne pripada noben dokument. Povezava med zadevami in dokumenti je vsebinsko pojasnjena v točki 7.5.

Za primer vzemimo, da v podatkovni bazi obstajajo naslednji zapisi za zadeve in dokumente, kot jih ponazarjata spodnji preglednici 4 in 5. Ena zadeva izmed petih ne vsebuje nobenega dokumenta (označena vrstica v preglednici). V spodnjih preglednicah so zaradi preglednosti navedeni le nekateri atributi tabele s slike 21, podatki so izmišljeni.

ID_ZADEVE	ID_DOSJEJA	ID_KLASIFIKACIJA	ID_ZAPOREDJE	DT_LETO	DT_EVIDENTIRANJE
1	1	10344	1	2016	10.01.2016
2	2	10321	1	2016	10.01.2016
3	3	10301	1	2016	10.01.2016
4	4	10390	1	2016	10.01.2016
5	5	10340	1	2016	10.01.2016

Preglednica 4: Testni primer – zapisi o zadevah

ID_DOKUMENTA	ID_ZADEVE	ID_ZAPOREDJE	ID_PRILOGA	DT_EVIDENTIRANJE	ID_VRSTA
1	1	1		10.01.2016	1
2	1	2		10.01.2016	5
3	2	1		10.01.2016	2
4	2	2		10.01.2016	3
5	2	2	1	10.01.2016	1
9	3	1		10.01.2016	2
10	3	2		10.01.2016	3
11	3	2	1	10.01.2016	1
12	4	1		10.01.2016	2
13	4	2		10.01.2016	3
14	4	2	1	10.01.2016	1

Preglednica 5: Testni primer – zapisi o dokumentih

Sledeči vnos bo ob proženju obdelave poskrbel za opozorilo uporabnika na to zadevo.

Uporabnik preko vmesnika *urejanje_poizvedb.jar* vnese naslednjo vsebino:

Šifra poizvedbe: npr. "*TEST1*"
Naziv poizvedbe: npr. "*Zadeve brez dokumentov*"
Opis: npr. "*Poizvedba nas opozori na zadeve brez dokumentov*"
Zahtevnost: izberemo npr. "*1 – časovno nezahtevno*"
Poslovno področje: izberemo npr. "*Splošni šifranti*"
Poizvedba: npr. "*select count(*) From ZADEVE a*
where not exists
(select 1 from DOKUMENTI b
where b.ID_ZADEVE = a.ID_ZADEVE)"
Velja od: trenutni datum v formatu *dd.mm.llll*
Velja do: brez vnosa

Uporabnik uporabi gumb **Shrani**. V prikazana polja na dnu vmesnika nato vnese naslednje:

Pričakovani rezultat: "*0*"
Obvestilo poslano, kadar: "*2 - če rezultat odstopa od pričakovanega*"
Email: poljuben elektronski naslov
Napotek za ukrepanje: npr. "*Poišči manjkajoče dokumente*"
Opis: brez vnosa

Velja od: trenutni datum v formatu *dd.mm.llll*
Velja do: prazno

Uporabnik uporabi gumb **Dodaj obveščanje.**

Preko vmesnika *zagon_obdelave.jar* uporabnik požene obdelavo (kot že omenjeno, bi se zagon v produkcijskem okolju moral izvajati samodejno po urniku). Uporabnik dobi rezultat kontrolne poizvedbe na elektronski naslov, ki ga je vnesel.

Elektronsko sporočilo bi bilo takšno:

Zadeva:
Rezultat nadzornih poizvedb 14.02.16 16:17:11,134000

Vsebina:
Pozdravljeni,

Obveščamo vas o rezultatu nadzornih poizvedb, na katere ste naročeni. To je samodejno generirano sporočilo. Prosimo, da nanj ne odgovarjate.

Rezultati v skladu s postavljenimi pravili v NADZOR in NADZOR_OBV:
ŠTEVILKA TRANSAKCIJE: 95 , PODROČJE: Splošni šifranti , UKREPANJE: Poišči manjkajoče dokumente , REZULTAT SQL: 1 , PRIČAKOVANI REZULTAT : 0 , KRATKA VSEBINA: Zadeve brez dokumentov

7.5 Vključitev rešitve v IS organizacije

V tej točki je podan predlog umestitve programske rešitve v informacijski sistem ZPIZ. V ta namen je rešitev iz točke 7.4 razširjena z dodatnimi funkcionalnostmi in z ustreznimi elementi za vključitev v obstoječ informacijski sistem ZPIZ. Predlagan sistem je v tej točki imenovan **Sistem za zaznavo napak**. Predlagan pa je tudi **Postopek za obravnavo napak**, znotraj katerega se uporabljajo funkcionalnosti predlaganega sistema. Znotraj Postopka za obravnavo napak je predlagan tudi **Postopek za popravek podatkov in odpravo vira napak.** Povezava med omenjenim sistemom in omenjenima postopkoma je prikazana na sliki 30.

Možno bi bilo razviti tudi samostojno programsko opremo za obravnavo napak, vendar bi za organizacijo, ki že razpolaga z določenimi infrastrukturnimi in programskimi rešitvami, to bilo neracionalno. Tako bi se podvajale določene funkcionalnosti, na primer programska oprema za delo na predhodno omenjenem postopku za obravnavo zaznanih napak. Ob uporabi omenjene obstoječe programske opreme (na sliki 33 je navedena kot Sistem za upravljanje z zadevami) lahko namreč uporabimo naslednje funkcionalnosti tega sistema:

- pregled statistike reševanja,
- pregled rešitev zadev,
- spremljanje poteka reševanja,
- dodeljevanje zadev v reševanje itd.

Zadeva je termin, ki ga predlaga Uredba o upravnem poslovanju (UUP) [95]. Uredba pojasnjuje, da je zadeva zbirka vseh dokumentov in prilog, ki se nanašajo na isto vsebinsko vprašanje ali nalogo. Primer takšne naloge je reševanje zahtevkov strank, prikazano na sliki 17. Zadeve pa predstavljajo tudi nekatere ostale naloge, ki jih rešujejo zaposleni. Primer takšne naloge oz. zadeve bi torej predstavljalo reševanje Postopka za obravnavo napak.

Zaradi predhodno zapisanega je bolj smiselna izdelava posameznih komponent in integracija v obstoječ informacijski sistem. Glavni pogoji oz. predpostavke integracije, ki so upoštevani v nadaljevanju so:

- uporaba dela obstoječega podatkovnega modela (prikazano na slikah 31 in 32);
    - za delo z zadevami,
    - za upravljanje s pooblastili in revizijsko sledjo,
    - za uporabo splošnih šifrantov,
- uporaba obstoječe podatkovne baze – DBMS (prikazano na sliki 33);
- uporaba obstoječih strežniških rešitev (prikazano na sliki 33);
    - uporaba obstoječega sistema BPM za vodenje postopkov in reševanje zadev,
    - uporaba strežnika WAS za izvajanje servisnih storitev,
    - uporaba obstoječe rešitve za proženje obdelav,

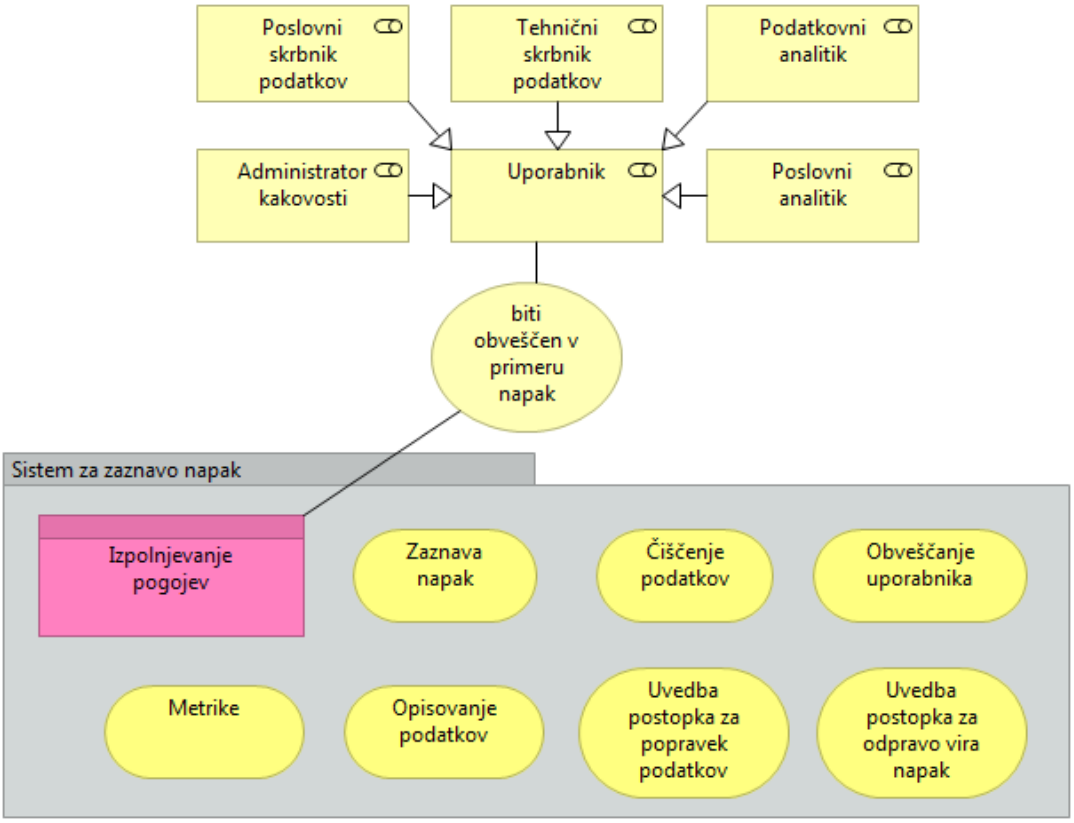
- uporaba obstoječega notranjega poštnega strežnika za sporočanje,

– uporaba obstoječega Sistema za upravljanje z zadevami (prikazano na sliki 33).

Predlagani Sistem za zaznavo napak se povezuje z naštetimi obstoječimi elementi informacijskega sistema. Spodnja slika 28 prikazuje model poslovnega produkta, ki ga predstavlja omenjeni sistem. Prikazuje vloge, ki sistem uporabljajo ter funkcionalnosti sistema. Funkcionalnosti prototipa (zaznavanje napak in obveščanje uporabnika) so za namen integracije v informacijski sistem razširjene s/z:

– čiščenjem podatkov,
– opisovanjem podatkov,
– uporabo za namen metrik,
– uvedbo Postopka za popravek podatkov,
– uvedbo Postopka za odpravo vira napak.

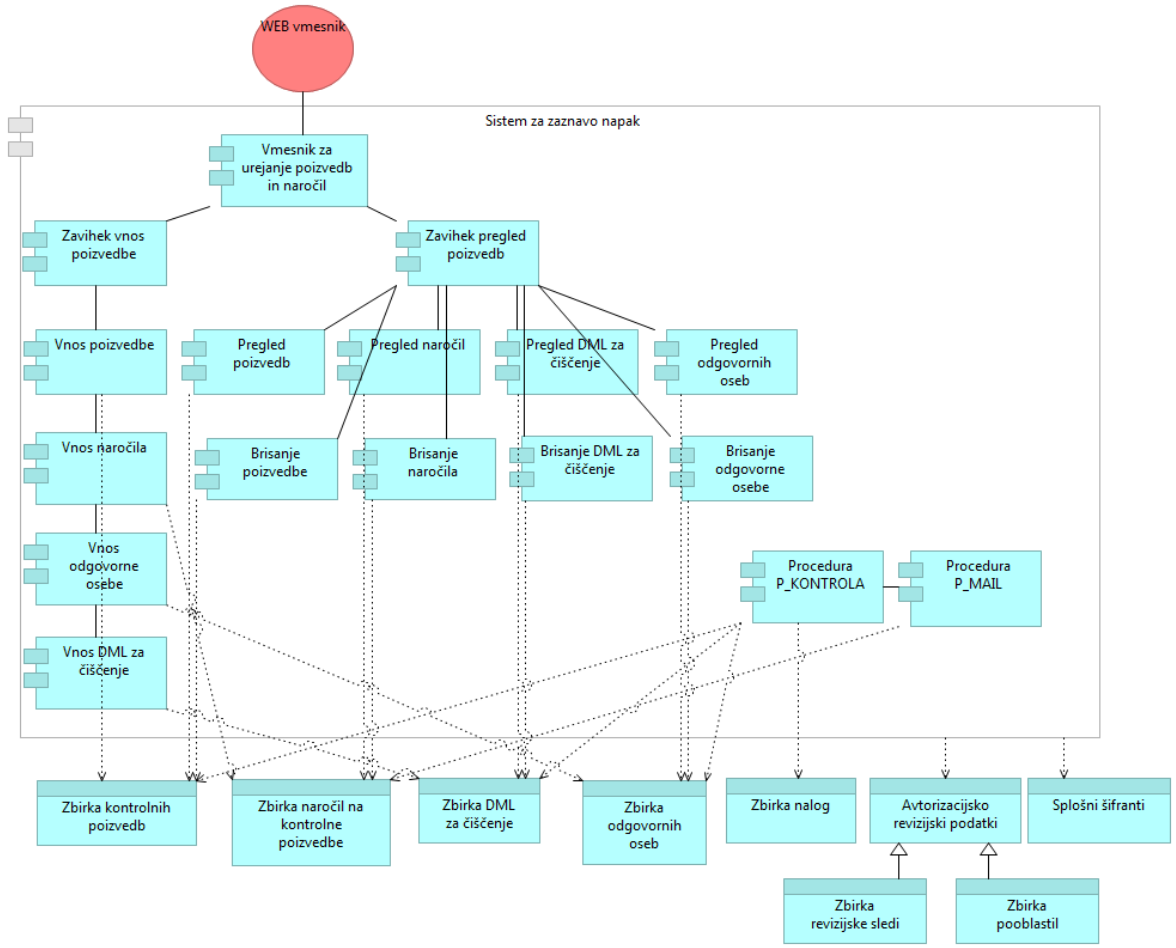
Pri tem sta lahko Postopek za popravek podatkov in Postopek za odpravo vira napak ločena postopka (kot je prikazano na sliki 28) ali pa združena v enega (kot je prikazano na sliki 30).



Slika 28: Model poslovnega produkta

Spodnja slika 29 prikazuje strukturo posameznih komponent aplikacije in njihove dostope do podatkovnih zbirk. Rešitev prototipa je na modelu razširjena z naslednjimi komponentami:

- *Vnos odgovorne osebe.* S pomočjo tega vmesnika za posamezno kontrolno poizvedbo vnesemo odgovorno osebo oz. uporabnika, ki v primeru zaznanih napak dobi nalogo za odpravo vira napak in odpravo napak v podatkih. Komponenta dostopa do zbirke odgovornih oseb, ki je v podatkovnem modelu na sliki 32 realizirana s tabelo NADZOR_ODGOVORNI.
- *Vnos DML za čiščenje.* V tem vmesniku za posamezno kontrolno poizvedbo vnesemo stavek DML ali stavek za zagon predhodno pripravljene procedure PL/SQL, ki se izvede v primeru zaznane napake. Komponenta dostopa do zbirke stavkov DML za čiščenje, ki je v podatkovnem modelu na sliki 32 realizirana s tabelo NADZOR_CISCENJE.
- *Pregled in brisanje odgovorne osebe.* S tem vmesnikom pregledujemo in odstranjujemo obstoječe odgovorne osebe za posamezne kontrolne poizvedbe.
- *Pregled in brisanje DML za čiščenje podatkov*. S tem vmesnikom pregledujemo in odstranjujemo obstoječe stavke za namen čiščenja podatkov.
- Vsebina procedure *P_KONTROLA* se glede na prototip razširi z uporabo zbirke stavkov za čiščenje podatkov in zbirke odgovornih oseb. V primeru zaznanih napak poleg obveščanja po elektronski pošti izvede stavke DML ali zažene predhodno izdelano proceduro PL/SQL ter izvede zapis v zbirko nalog za namen uvedbe Postopka za popravek podatkov in odpravo vira napak, kot je označen na sliki 30.
- Uporaba zbirke nalog, ki je na sliki 32 realizirana s tabelo DODELITEV_NALOG.
- Dostop do ostalih obstoječih zbirk informacijskega sistema: splošnih šifrantov in do zbirke avtorizacijskih in revizijskih podatkov za namen zapisovanja revizijske sledi in preverjanja pooblastil za uporabo Sistema za zaznavo napak.



Slika 29: Model strukture aplikacije

Spodnja slika 30 prikazuje predlagani **Postopek za obravnavo napak**, ki bi se lahko uporabljal ob zaznani napaki v podatkih. Postopek je opisan spodaj, njegovi glavni gradniki postopka so:

- tehnični postopek, ki se izvaja samodejno znotraj Sistema za zaznavo napak;
- korak za izdelavo Postopka za popravek podatkov in odpravo vira napak;
- Postopek za popravek podatkov in odpravo vira napak.

Za proženje funkcionalnosti zaznavanja napak znotraj Sistema za zaznavanje napak se uporabi obstoječa funkcionalnost informacijskega sistema za proženje obdelav po urniku, ki je na spodnji sliki ponazorjena kot storitev samodejnega proženja.

Omenjeni tehnični postopek se izvaja znotraj procedur PL/SQL Sistema za zaznavo napak in vsebuje naslednje funkcionalnosti: sistem najprej poišče nabor kontrolnih poizvedb, ki jih mora izvesti (prebere jih iz tabele NADZOR). Izvede vsako posebej, v primeru zaznanega odstopanja dejanskega rezultata od pričakovanega rezultata sledi nadaljnji potek. Sistem obvesti enega ali več uporabnikov, ki so naročeni na obveščanje za to kontrolno poizvedbo

(seznam naročenih uporabnikov prebere iz tabele NADZOR_OBV). V primeru, ko je za kontrolno poizvedbo definirana vsebina v tabeli NADZOR_CISCENJE, sledi čiščenje podatkov. V tem primeru se izvede en ali več stavkov DML ali stavkov za zagon predhodno pripravljene procedure PL/SQL. Primer scenarija, ko se uporabi stavek DML za čiščenje podatkov, je lahko naslednji; če sistem zazna razliko v datumu rojstva in datumskem delu EMŠO in je organizacija prepričana v pravilnost EMŠO, lahko za zaznane napake izvede naslednji stavek DML za čiščenje podatkov:

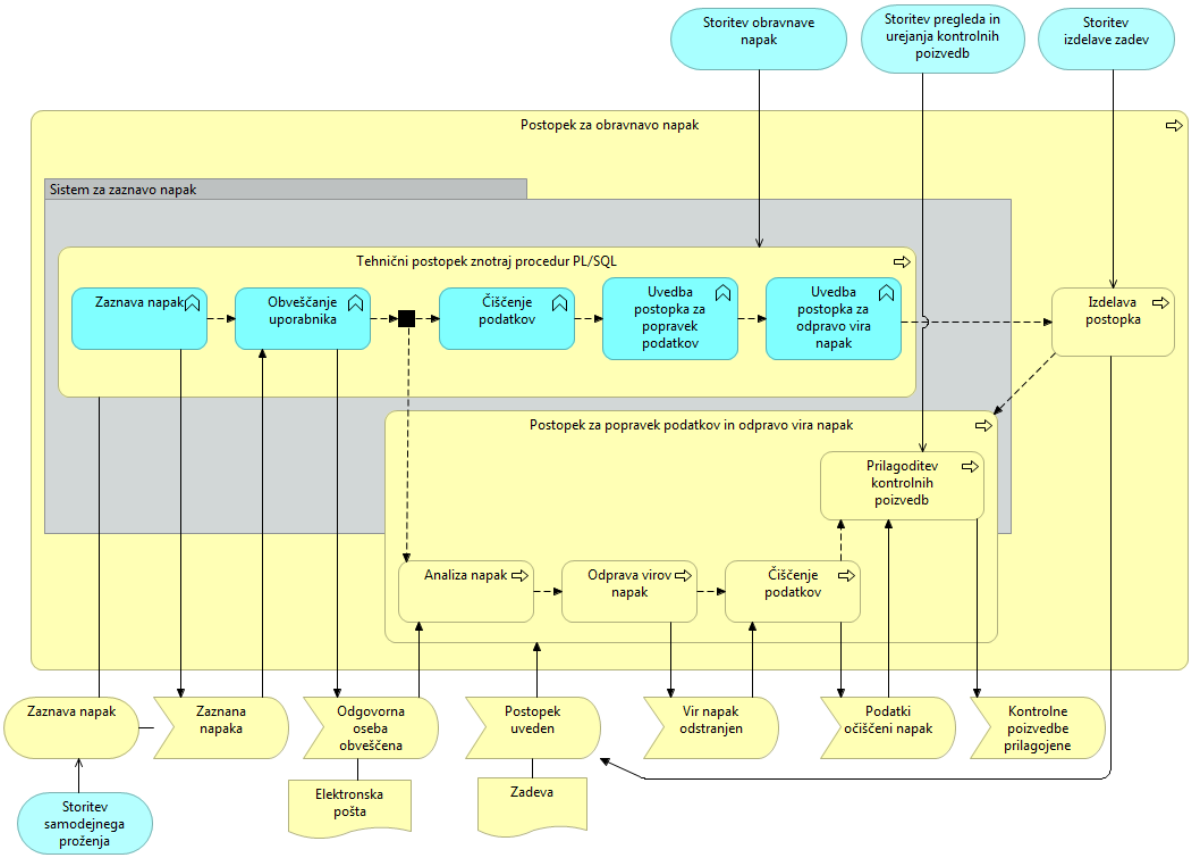
```
update ZAVAROVANCI
set DT_DAT_ROJ =
        to_date(        substr(TX_EMSO,1,4) ||
                        case when substr(TX_EMSO,5,1) < 5  then '2' else '1' end ||
                        substr(TX_EMSO,5,3) ,'ddmmyyyy'  )
where ID_ZAVAROVANCA = :identifikator osebe z zaznano napako ;
```

Sledi uvedba Postopka za popravek podatkov in odpravo vira napak (označen na sliki 30). V ta namen se uporabi vnos v NADZOR_ODGOVORNI, kjer preberemo, komu dodeliti zadevo za vodenje postopka. Rezultat tega koraka je eden ali več zapisov v tabeli DODELITEV_NALOG, ki predstavlja obstoječo tabelo informacijskega sistema.

Hkrati s čiščenjem podatkov, ki ga izvaja sistem samodejno glede na definirane stavke DML v tabeli NADZOR_CISCENJE, lahko uporabniki, ki so prejeli elektronsko sporočilo, pričnejo z analizo zaznane napake. Zadeva za Postopek za popravek podatkov in odpravo vira napak pa jim je formalno dodeljena ob izvedbi koraka za izdelavo postopka. Ta korak je v obstoječem informacijskem sistemu realiziran z redno obdelavo, ki se vrši v rednih časovnih periodah in glede na čakajoče naloge v tabeli DODELITEV_NALOG izdela zadeve za vodenje postopkov.

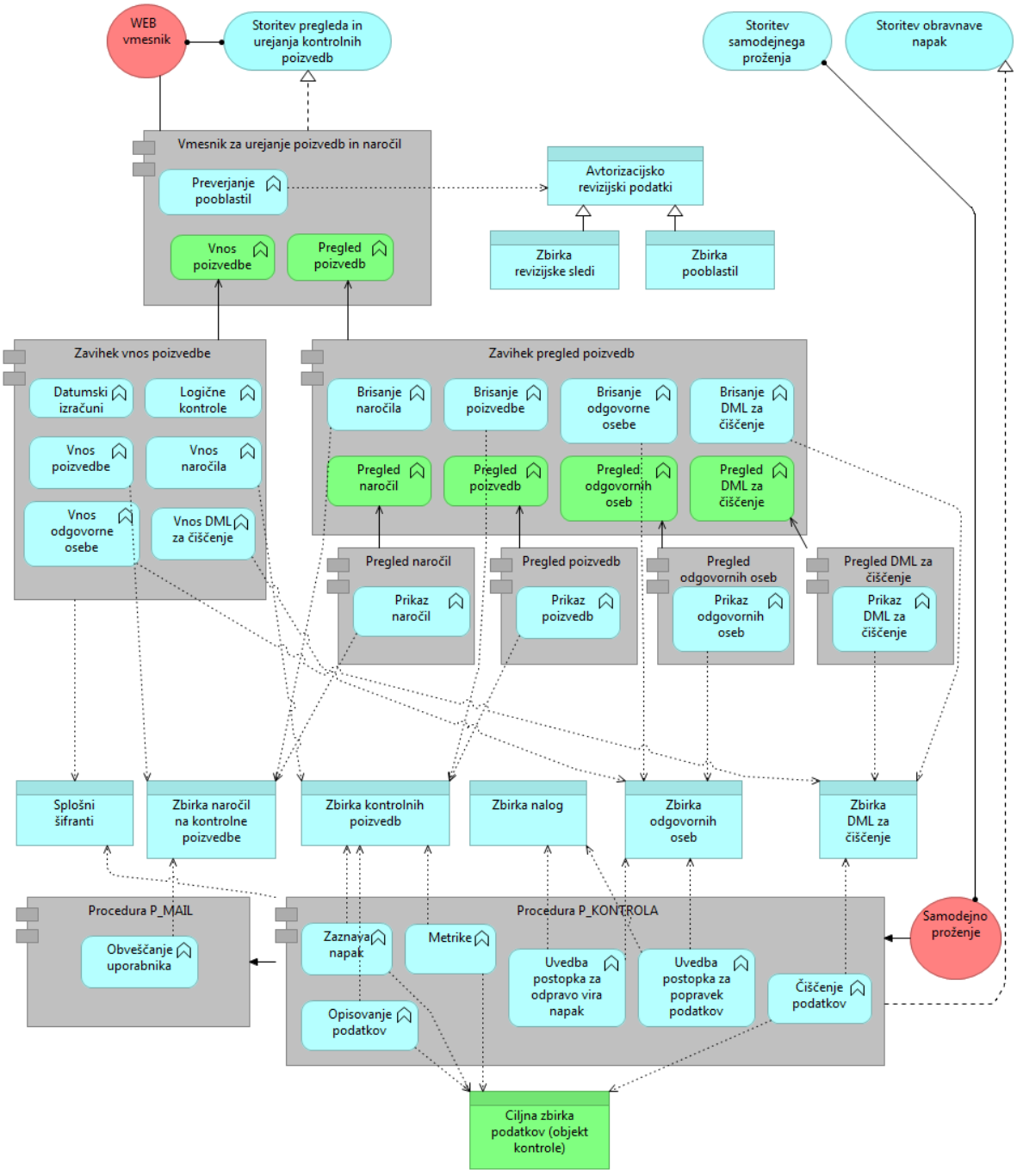
Uporabniki torej analizirajo napake, odpravijo vir napak, na primer s konkretnimi popravki v kodi programske opreme, vzpostavitvijo novih kontrol na vnosih podatkov, ustrezno dopolnitvijo podatkovnega modela ali baznih objektov, navodili za uporabnike itd. Ko je vir napak odpravljen, sledi odprava zaznanih napak v podatkih. Ta korak dopolnjuje istoimenski korak znotraj Sistema za zaznavo napak. Za določene vrste napak je namreč možno vnaprej predvideti ustrezne stavke DML (npr. zgoraj opisani primer), določene pa zahtevajo ročni poseg uporabnikov, predvsem pri napakah, kjer sistem ne more enostavno pridobiti pravilnih vrednosti. Po potrebi sledi korak za prilagoditev kontrolnih poizvedb glede na predhodno izvedeno analizo, pričakovanja o nadaljnem pojavljanju napake in pridobljena nova znanja.

Na modelu je glede na opisan postopek prikazan tudi potek poslovnih dogodkov, ki si sledijo v primeru zaznane napake.



Slika 30: Model procesa za obravnavo napak

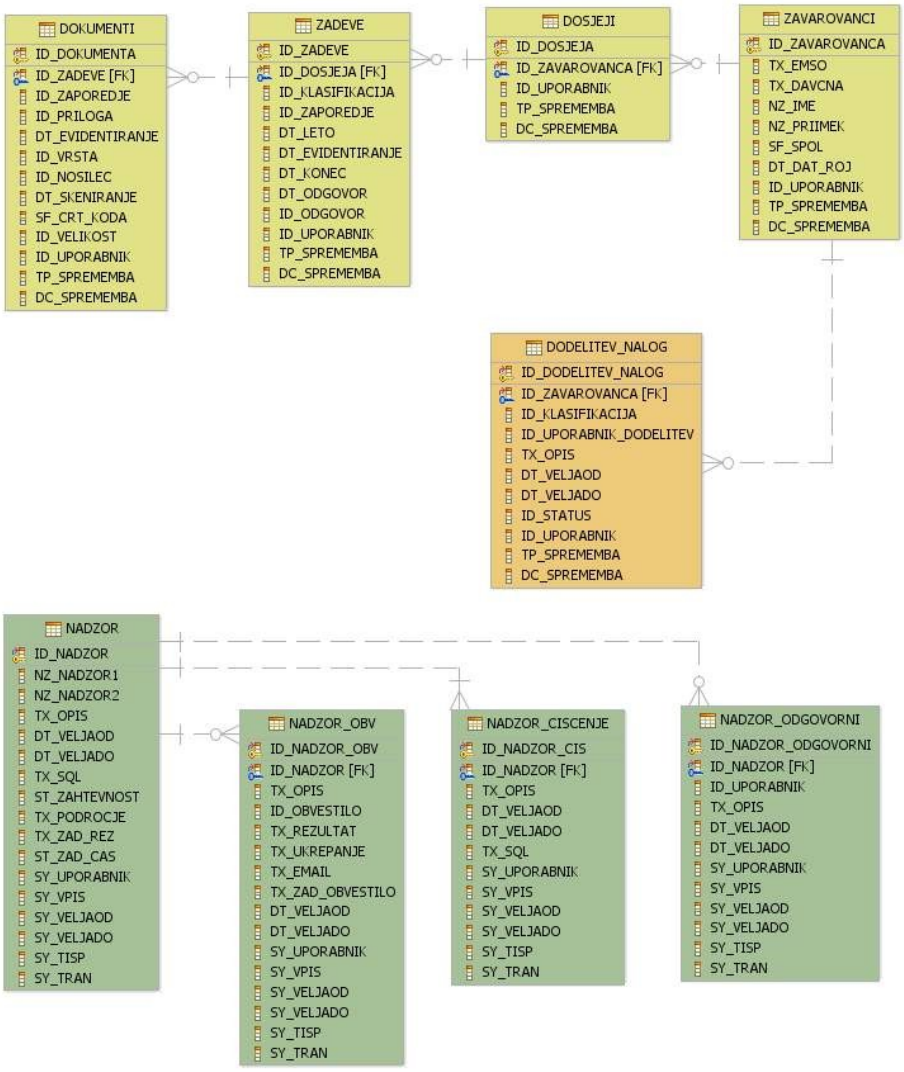
Spodnja slika 31 prikazuje predlagani Sistem za zaznavo napak na aplikacijskem nivoju, kjer so razvidne posamezne funkcionalnosti sistema znotraj posameznih komponent sistema, njihove povezave in dostopi do podatkovnih zbirk. Posamezne komponente so opisane v točki 7.4 ter v opisu slike 29.



Slika 31: Model aplikacijskega nivoja

Spodnji podatkovni model na sliki 32 je razširitev podatkovnega modela za podporo prototipa s slike 21. Za namen dodatnih funkcionalnosti vsebuje dodatne tabele:

- NADZOR_CISCENJE, ki se uporabi za shranjevanje stavkov DML ali stavkov za zagon poljubnih procedur PL/SQL, ki se zaženejo v primeru zaznanih napak posameznih kontrolnih poizvedb, shranjenih v tabeli NADZOR.
- NADZOR_ODGOVORNI, ki se uporabi za shranjevanje odgovornih oseb, ki v primeru zaznanih napak dobijo nalogo za odpravo napak v podatkih in odpravo vira napak.
- DODELITEV_NALOG ponazarja tabelo obstoječega informacijskega sistema, kjer se zbirajo naloge, ki so kasneje dodeljene v delo uporabnikom v sistemu BPM.
- Ostale tabele predstavljajo del obstoječega podatkovnega modela za podporo sistema za upravljanje z zadevami in se v predlaganem sistemu uporabijo za reševanje zadev, izdelanih na podlagi nalog iz tabele DODELITEV_NALOG. Imena teh obstoječih objektov (tabel in njihovih atributov) so zaradi varnosti spremenjena in ne odražajo dejanskega stanja. Objekti so za namen preglednosti poenostavljeni.

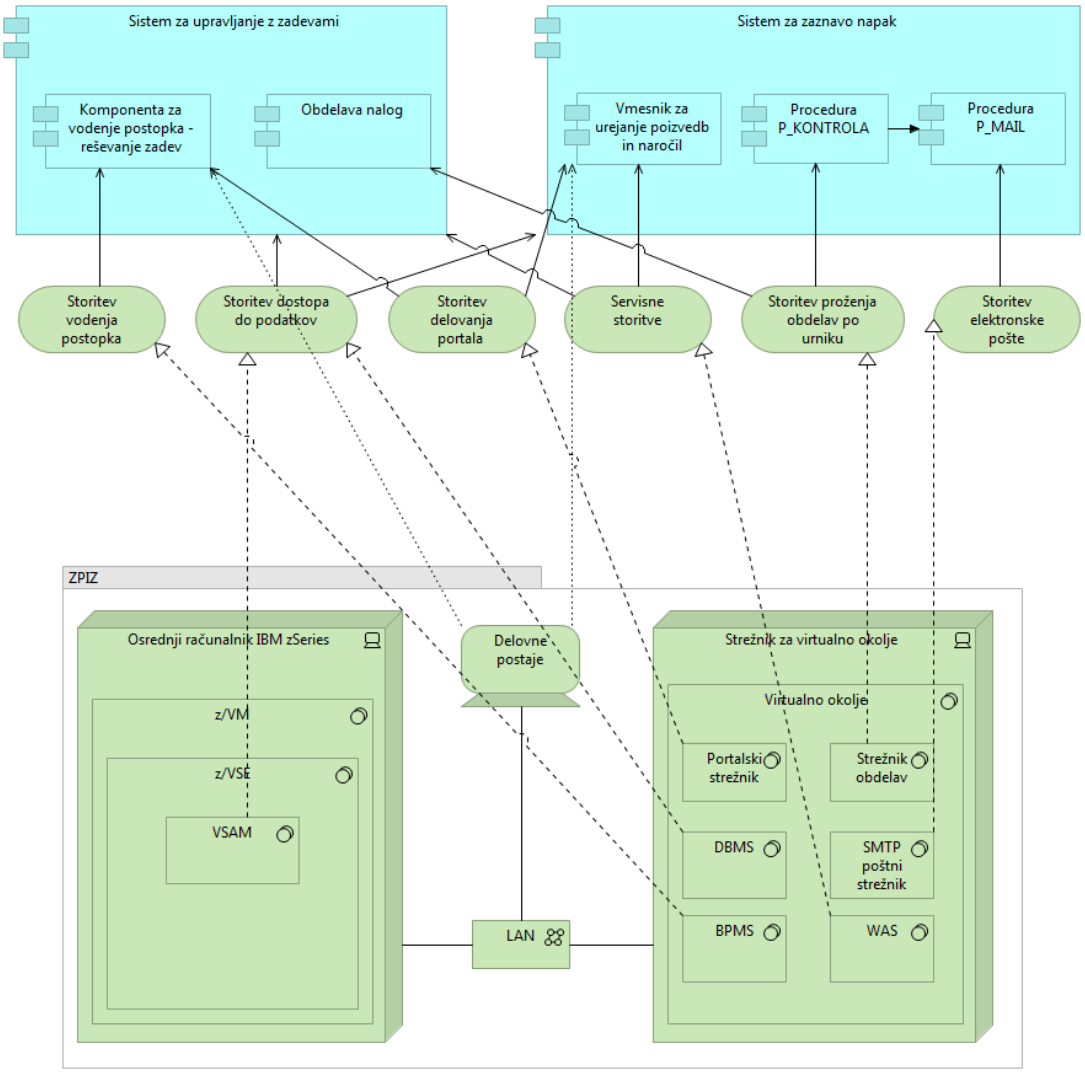


Slika 32: Podatkovni model

Spodnja slika 33 prikazuje možno namestitev komponent Sistema za zaznavo napak v informacijsko infrastrukturo ZPIZ. Sistem za upravljanje z zadevami predstavlja obstoječ sistem in se uporablja za:

- izdelavo zadev za vodenje postopkov na podlagi zapisov tabele DODELITEV_NALOG, kamor naloge zavede predlagani Sistem za zaznavo napak;
- vodenje postopkov in reševanje zadev. Tako poteka tudi delo na osnovnem poslovnem procesu ZPIZ, predstavljenem na sliki 17.

Razlika glede na prototip s slike 20 je tudi ta, da se za pošiljanje elektronske pošte uporabi notranji poštni strežnik. To in pa možnost uporabe orodja Oracle Wallet Manager poenostavi realizacijo komponente za izdelavo in pošiljanje elektronske pošte, procedure *P_MAIL*. V tem primeru se lahko namreč uporabi notranje varnostne nastavitve, enostavnejše sestavljanje elektronskih sporočil ter enostavnejšo komunikacijo procedure in poštnega strežnika.



Slika 33: Tehnološka infrastruktura in model namestitve

7.6 Upoštevana vodila in smernice

Pri snovanju programske rešitve in predlogu umestitve v obstoječi informacijski sistem so upoštevane smernice in vodila, ki so bili predhodno predstavljeni v točki 6.2.2. Spodnje preglednice prikazujejo ta vodila in določajo posamezne komponente predlaganega sistema in postopka (opisanega v točki 7.5), ki odgovarjajo posameznemu vodilu.

Vodila dveh D, dveh P in dveh R, ki jih predlaga Geiger [20], so v Sistemu za zaznavo napak in Postopku za obravnavo napak upoštevane na način, kot jih prikazuje preglednica 6.

<table>
<tr><th>Vodilo</th><th>Splošni pomen</th><th>Realizacija</th></tr>
<tr><td>Odkrij
(ang. detect).</td><td>Izvedba analize podatkov</td><td>Funkcionalnost zaznave napak</td></tr>
<tr><td>Odvrni
(ang. deter).</td><td rowspan="2">Odprava nepravilnosti v postopkih in programski opremi, ki napake povzroča, ter vgradnja dodatnih omejitev za preprečevanje napak.</td><td rowspan="2">Funkcionalnost uvedbe Postopka za odpravo vira napak</td></tr>
<tr><td>Preprečuj
(ang. prevent).</td></tr>
<tr><td>Pripravi
(ang. prepare).</td><td>Priprava ustreznih postopkov, vlog, odgovornosti in dolžnosti na nivoju organizacije</td><td>Funkcionalnost vnosa kontrolnih poizvedb, naročil, DML in odgovornih oseb</td></tr>
<tr><td>Reagiraj
(ang. respond).</td><td rowspan="2">Izvedba čiščenja podatkov ter obveščanje virov napak</td><td>Postopek obravnave napak</td></tr>
<tr><td>Povrni
(ang. recover).</td><td>Funkcionalnost čiščenja podatkov, funkcionalnost uvedbe Postopka za popravek podatkov</td></tr>
</table>

Preglednica 6: Upoštevanje vodil dveh D, P in R

Osnovni vodili za zagotavljanje kakovosti podatkov, ki jih navajajo Cappiello, Francalanci in Pernici [8], so upoštevani z navedenim v spodnji preglednici 7.

Vodilo	**Realizacija**
Neprekinjena kontrola vrednosti podatkov, shranjenih v podatkovni bazi	Funkcionalnost zaznave napak skupaj z uporabo obstoječe storitve samodejnega proženja
Popravek v primeru napak	Funkcionalnost uvedbe postopka za popravek napak

Preglednica 7: Upoštevanje vodil [8]

Predlog Sistema za zaznavo napak je tehnično zasnovan za visoko umestitev v zrelostni model po modelu COBIT [82] (npr. stopnja štiri). Ta cilj bi lahko bil dosežen s preglednostjo vodenja postopka obravnave napak, popravka podatkov in odprave vira napak preko BPM, rezultati so v tem primeru tudi merljivi preko obstoječih orodij za statistiko. V postopku za obravnavo napak, predstavljenem v točki 7.5, je s predvidenim korakom za prilagoditev kontrolnih poizvedb poskrbljeno tudi za izboljševanje in optimizacijo zaznavanja posameznih napak. Za izpolnjevanje takšnega zrelostnega nivoja pa je rešitev treba tudi pravilno implementirati z organizacijskega vidika – kar pomeni določitev ustreznih pooblastil, vlog, zadolžitev itd.

Spodnja preglednica 8 je pregled vodil, ki jih navaja Chapman [9] (podrobnejša razlaga posameznih vodil se nahaja v točki 6.2.2). Nekatera vodila so organizacijske narave in jih nekoliko težje razumemo v tehnični luči kot predhodno navedena vodila.

Vodilo	**Realizacija**
Načrtovanje (razvoj vizije, politike in strategije)	Uvedba predlaganega Postopka za obravnavo napak
Primerna organizacija podatkov izboljšuje učinkovitost.	Oblikovanje ustreznih kontrolnih poizvedb po vodilu "deli in vladaj" – delitev predvidenih napak na več manjših, lažje obvladljivih, s tem je tudi definiranje stavkov DML za čiščenje podatkov enostavnejše. Npr. iskanje napak v naslovih po različnih državah, kjer veljajo različna pravila.
Preprečevanje je boljše kot kasnejše čiščenje.	Sistem zaznave napak je namenjen reaktivnemu pristopu DQM, zato tega vodila ne moremo uporabiti.
Primerna delitev odgovornosti	Pretežno organizacijsko vodilo. Realizacija je predvidena z uporabo zbirke odgovornih oseb.
Sodelovanje izboljšuje učinkovitost.	Organizacijsko vodilo
Primerna postavitev prioritet	Sistem med kontrolnimi poizvedbami sicer ne dela razlik glede prioritet, imamo pa možnost vključevanja in izključevanja posameznih poizvedb, s tem pa lahko posredno določamo prioritete.
Postavitev ciljev in metrik	Funkcionalnost metrike – sistem lahko uporabimo tudi za izvajanje metrik, s primerno oblikovanimi kontrolnimi poizvedbami in izbiro pošiljanja obvestil ne glede na rezultat.
Izogibanje podvojenemu čiščenju (uporaba ustreznega podatkovnega modela za spremljanje preteklih akcij)	Za izpolnitev tega vodila bi uporabili obstoječo prakso arhivskih tabel, kjer je to potrebno.
Zagotavljanje povratnih informacij	Funkcionalnost obveščanja uporabnikov v fazi zaznave napak. Kasneje obveščanje ni predvideno, rešitev je razvidna iz obstoječega Sistema za upravljanje z zadevami.
Izobraževanje	Organizacijsko vodilo
Zadolžitve, preglednost, sledljivost spremembam	Pretežno organizacijsko vodilo. Preglednost je dosežena z uporabo obstoječega Sistema za upravljanje z zadevami.
Dokumentiranje	Za izpolnitev tega vodila bi uporabili obstoječo prakso arhivskih tabel.

Preglednica 8: Upoštevanje vodil [9]

Postopek za obravnavo napak lahko ponazorimo tudi z Demingovim krogom, prikazanim na sliki 11. Glede na razlago Demingovega kroga v točki 5.1.2 bi posamezne elemente postopka razvrstili, kot prikazuje preglednica 9.

Korak	**Del Postopka za obravnavo napak**
Načrtuj.	Začetni vnos in prilagoditev kontrolnih poizvedb
Naredi.	Izvajanje funkcionalnosti zaznave napak
Preveri.	Izvajanje funkcionalnosti zaznave napak – primerjava dejanskih in pričakovanih rezultatov
Ukrepaj.	Funkcionalnosti obveščanja uporabnika, čiščenja podatkov, uvedbe postopkov za popravek podatkov in odpravo vira napak, podproces izdelave napak, postopka, analiza odprave virov napak in ročni del čiščenja podatkov

Preglednica 9: Demingov krog in Postopek za obravnavo napak

8. Zaključek

Glede na pregledano literaturo in vire lahko trdim, da vsaka organizacija, ki v svojih poslovnih postopkih uporablja podatke, potrebuje ustrezni nivo kakovosti podatkov. To je še posebej pomembno zaradi teže posledic nepravilnosti, ki jih nosijo vse ravni organizacije – na najvišji ravni organizacij se kažejo kot poslovna škoda, na ravni zaposlenih kot nezadovoljstvo pri delu, zunaj organizacije pa kot nezadovoljstvo strank in zmanjšanje ugleda.

Zagotavljanje ustrezne ravni kakovosti podatkov terja celostni pristop organizacije k izoblikovanju postopka za upravljanje kakovosti podatkov ali DQM. Celostni pristop pri tem pomeni, da je postopek pozitivno sprejet na nivoju celotne organizacije, preko vseh organizacijskih enot, in vključuje vse ravni zagotavljanja kakovosti, torej od samih tehničnih postopkov do ustrezno zasnovanih postopkov, ki uporabljajo in spreminjajo podatke.

Medtem ko mora poslovne procese za zagotavljanje kakovosti podatkov zaradi specifičnih lastnosti vsake organizacije zagotoviti vsaka organizacija sama z ozirom na lastno strategijo, vizijo in cilje, so na trgu na voljo posamezna orodja za pomoč pri posameznih konkretnih točkah zagotavljanja kakovosti podatkov. Na voljo so tudi ogrodja, na primer [41], ki organizacijam nudijo osnovo za uvedbo postopka.

V nalogi sem kot prispevek k omenjenem področju predstavil predlog orodja oz. Sistema za zaznavanje napak, ki odgovarja na več problemskih domen DQM, in možno uporabo prikazal na predlogu integracije v obstoječ informacijski sistem ZPIZ ter predlagal Postopek za obravnavo napak (slika 30). Upošteval sem smernice in principe, ki jih predlaga literatura [8, 9, 20]. Vodilo pri zasnovi in predlogu integracije je bilo tudi to, da sistem napake zaznava (in po možnosti odpravi) samodejno in posledično v največji možni meri zmanjša posledice zaradi ponavljanja istovrstnih napak. V primeru večjih organizacij, kjer se določen podatek spreminja na podlagi več virov, namreč obstaja nevarnost, da se napaka sicer zazna in odpravi, vendar ne na vseh virih. Predlagana rešitev naslavlja tudi takšne težave.

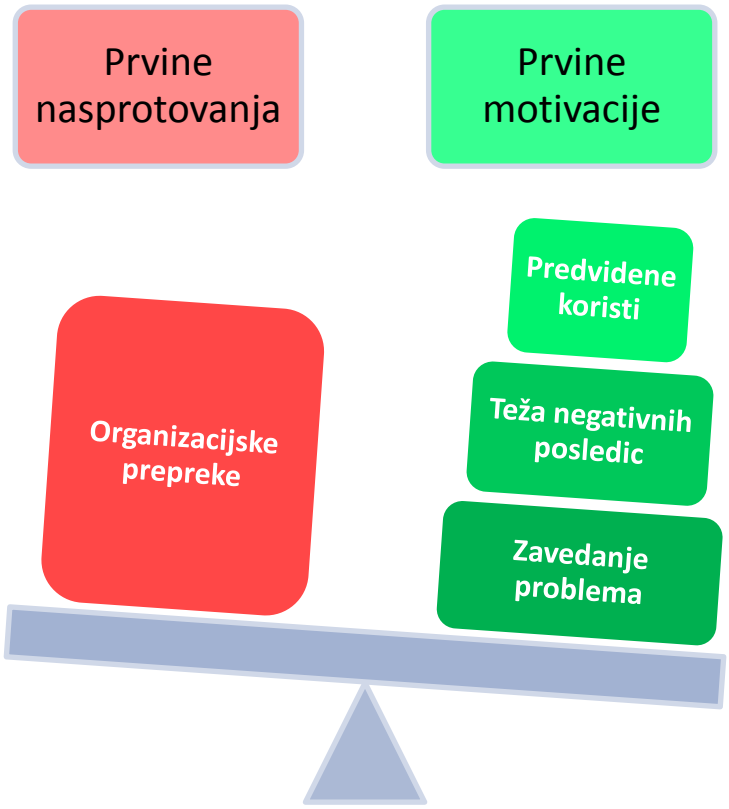
Prototip orodja je bil na omejeni testni podatkovni bazi preizkušen in je deloval v skladu s pričakovanji in opisom v točki 7.4. Delovanje celotne predlagane razširjene rešitve pa bi lahko ocenil po dejanski integraciji, pričakujem pa, da bi bil doprinos h kakovosti podatkov ob vseh v nadaljevanju podanih predpostavkah pozitiven. Omenjene predpostavke so zelo podobne kot pri vsakem uvajanju novih sistemov na nivoju organizacije:

- sprejem pri vodstvu,
- sprejem pri uporabnikih,
- doslednost uporabe za predviden namen,

- ustrezno določene vloge in odgovornosti,
- nadzor vodstva pri odpravljanju napak in virov napak.

Obseg literature kaže na resnost problema in njegovih posledic, množica različnih vzrokov in dejstvo, da je glede na literaturo problem aktualen in prisoten skozi več kot dve desetletji, pa kažeta na težavnost odprave problema oz. zagotavljanja ustrezne ravni kakovosti podatkov.

Odločanje organizacij o vpeljavi postopka bi lahko ponazorili s spodnjo sliko 34. V primeru, da prvine motivacije prevladajo nad prvinami nasprotovanja, bi se organizacija morala odločiti za vpeljavo postopka DQM. Pri tem je zavedanje problema temeljni predpogoj za vpeljavo. Organizacijske prepreke pa vključujejo finančna sredstva, ki so na voljo, število zaposlenih za podporo pri vpeljavi, podporo poslovnega vodstva itd.



Slika 34: Odločanje o vpeljavi postopka za upravljanje kakovosti podatkov

Literatura in viri

Literatura

[1] D. Ballou in ostali, "Modeling information manufacturing systems to determine information product quality," *Management Science*, vol. 44, št. 4, str. 462–484, april 1998.

[2] P. Barnaghi, A. Sheth in C. Henson, "From data to actionable knowledge: big data challenges in the web of things," *IEEE Intelligent Systems*, vol. 28, št. 6, str. 6–11, 2013.

[3] C. Batini, M. Lenzerini in S. B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys (CSUR),* vol. 18, št. 4, str. 323–364, december 1986.

[4] C. Batini in M. Scannapieco, *Data quality: Concepts, Methodologies and Techniques*. New York: Springer, 2006.

[5] M. Betts, "Dirty data: Inaccurate data can ruin supply chains projects," *Computerworld,* vol. 35, št. 51, str. 42, december 2001.

[6] J. E. Boritz, "IS practioners' view on core concepts od information integrity," *International Journal of Accounting Information Systems,* vol. 6, št. 4., str. 260–279, Elsevier, december 2005.

[7] P. A. Burrough in R. A. McDonnell, *Principals of Geographical Information Systems*. Oxford, UK: Oxford University Press, 1998.

[8] C. Cappiello, C. Francalanci in B. Pernici, "A Self-monitoring System to Satisfy Data Quality Requirements," v zborniku *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31–November 4 2005,* Lecture Notes in Computer Science, vol. 3761, R. Meersman, Z. Tari in ostali, ur. Berlin; Heidelberg: Springer, 2005, Part II, str. 1535–1552.

[9] A. D. Chapman. (2005). *Principles and Methods of Data Cleaning – Primary Species and Species-Occurrence Data* [Online]. Report for the Global Biodiversity Information Facility. Copenhagen: Global Biodiversity Information Facility GBIF.
Dostopno na: http://www.gbif.org/orc/?doc_id=1262 .

[10] A. D. Chapman. (2005). *Principles of Data Quality* (verzija 1) [Online]. Report for the Global Biodiversity Information Facility. Copenhagen: Global Biodiversity Information Facility GBIF. Dostopno na: http://www.gbif.org/orc/?doc_id=1229 .

[11] A. D. Chapman, "Quality Control and Validation of Point-Sourced Environmental Resource Data," v *Spatial accuracy assessment: Land information uncertainty in natural resources*, K. Lowell, ur., A. Jaton, ur. Chelsea, Michigan: Ann Arbor Press, 1999, str. 409–418.

[12] G. Cong in ostali, "Improving data quality: Consistency and accuracy," v zborniku *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB 2007), September 23–27, 2007, Vienna, Austria*. VLDB Endowment, 2007, str. 315–326.

[13] S. De in ostali, "BayesWipe: A Multimodal System for Data Cleaning and Consistent Query Answering on Structured BigData," v zborniku *Proceedings – 2014 IEEE International Conference on Big Data, IEEE Big Data 2014, Washington, United States, 27–30 October*. Institute of Electrical and Electronics Engineers Inc., 2015, str. 15–24.

[14] A. Deutsch in ostali, "Xml-ql: A query language for xml," v *Proceedings of the 8th International World Wide Web Conference, May 11–14, 1999, Toronto, Canada* [Online]. Dostopno na: http://www8.org/w8-papers/1c-xml/query/query.html .

[15] C. Falge, B. Otto in H. Österle, "Towards a Strategy Design Method for Corporate Data Quality Management," v zborniku *Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013), 27th February–1st March, Leipzig, Germany,* R. Alt, ur. B. Franczyk, ur. Leipzig: Universität Leipzig, 2013, vol. 1, str. 801–815.

[16] E. B. Fernandez, R. C. Summers in C. Wood, *Database Security and Integrity*. Reading, Maassachusetts: Addison-Wesley, 1981.

[17] A. Ferriyan in J. E. Istiyanto, "Data Center Governance Information Security Compliance Assessment Based on the Cobit Framework," *International Journal of Advanced Computer Science and Applications,* vol. 6, št. 2, str. 34–36, 2015.

[18] C. W. Fisher in B. R. Kingma, "Criticality of data quality as exemplified in two disasters," *Information & Management,* vol. 39, št. 2, str. 109–116, Elsevier, december 2001.

[19] J. G. Geiger, "Data Quality Management: The Most Critical Initiative You Can Implement," v zborniku *SUGI 29 Proceedings: SAS Users Group International Conference, May 9–12, 2004, Palais Des Congrès de Montréal, Montréal, Canada*. Cary, North Carolina: SAS Institute, 2004, Paper 098-29.

[20] E. Gelbstein, "Data Integrity – Information Security's Poor Relation," *Isaca Journal,* št. 6, str. 20–31, 2011.

[21] J. A. Ghaeb, M.A. Smadi in J. Chebil, "A high performance data integrity assurance based on the determinant technique," *Future Generation Computer Systems,* vol. 27, št. 5, str. 614–619, Elsevier, maj 2011.

[22] D. Greenfield. (6. december 2007). Standards for IT Governance: ITIL, COBIT, and ISO 17799 Provide a Blueprint for Managing IT Services. *InformationWeek* [Online]. Dostopno na: http://www.informationweek.com/standards-for-it-governance/d/d-id/1062203 .

[23] D. Hvala, "ITIL – del rešitve ali del problema?," *MonitorPro,* št. 1 (pomlad), 2012.

[24] V. Kashyap in A.P. Sheth, "Semantic and Schematic Similarities between Database Objects: A Context-Based Approach," *The VLDB Journal,* vol. 5, št. 4, str. 276–304, december 1996.

[25] M. Krisper in ostali, *Enotna metodologija razvoja informacijskih sistemov EMRIS, zvezek 3, Strukturni razvoj*, 2. Izdaja. Ljubljana: Vlada Republike Slovenije, Center Vlade RS za informatiko, 2004.

[26] M. Krumenaker, G. Bukhbinder in X. Yang, "SAS Data Quality – Cleanse: Techniques for Merge/Purge on Very Large Datasets," v zborniku *SUGI 29 Proceedings: SAS Users Group International Conference, May 9–12, 2004, Palais Des Congrès de Montréal, Montréal, Canada*. Cary, North Carolina: SAS Institute, 2004, Paper 014-29.

[27] T. Lahdenmäki, *Relational Database Index Design and the Optimizers*. Hoboken, New Jersey: John Wiley and Sons, 2005.

[28] Y. W. Lee in ostali, *Journey to Data Quality*. Cambridge, Massachusetts: MIT Press, 2006.

[29] D. Loshin. (1. maj 2005). Developing Information Quality Metrics. *Information management* [Online]. Dostopno na: http://www.information-management.com/issues/20050501/1026061-1.html .

[30] D. Loshin, *Enterprise knowledge management: the data quality approach*. San Diego: Morgan Kaufmann, 2001.

[31] D. Li, J. Zhang in H. Wu, "Spatial data quality and beyond," *International Journal of Geographical Information Science*, vol. 26, št. 12, str. 2277–2290, Taylor & Francis, september 2012.

[32] T. W. Ling, C. H. Goh in M. L. Lee, "Extending classical functional dependencies for physical database design," *Information and Software Technology,* vol. 38, št. 9, str. 601–608, Elsevier, 1996.

[33] M. Ma, P. Wang in C. H. Chu, "Datamanagement for Internet of Things: Challenges, Approaches and Opportunities," v zborniku *Proceedings of the Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings / CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, 20–23 August 2013, Beijing, China*. Washington, DC, USA: IEEE Computer Society, 2013, str. 1144–1151.

[34] J. I. Maletic in A. Marcus. Data Cleansing: Beyond Integrity Analysis. V zborniku *Proceedings of the Conference on Information Quality (IQ2000), 20–22 October 2000, Boston: Massachusetts Institute of Technology, Massachusetts* [Online]. Str. 200–209. Dostopno na: http://www.sdml.info/papers/IQ2000.pdf .

[35] M. Mihelčič, *Poslovne funkcije,* 5. popravljena in dopolnjena izdaja. Ljubljana: Fakulteta za računalništvo in informatiko, 2008.

[36] N. Mishra, C. C. Lin in H. T. Chang, "A Cognitive Adopted Framework for IoT Big-Data Management and Knowledge Discovery Prospective," *International Journal of Distributed Sensor Networks*, vol. 2015, ID članka 718390, 12 strani, Hindawi, 2015.

[37] G. Moerkotte in P.C. Lockemann, "Reactive consistency control in deductive databases," *ACM Transactions on Database Systems*, vol. 16, št. 4, str. 670–702, december 1991.

[38] P. Nastase, F. Nastase in C. Ionescu, "Challenges generated by the implementation of the IT standards COBIT 4.1, ITIL V3, and ISO/IEC 27002 in enterprises," *Economic Computation & Economic Cybernetics Studies & Research,* vol 43, št.3, str. 5–20, julij 2009.

[39] B. Nightingale. (31. maj 2007). ITIL and Data Quality: A Familiar Partnership. *Information management* [Online]. Dostopno na: http://www.information-management.com/news/columns/-1083375-1.html .

[40] C. A. O'Reilly III, "Variations in decision makers' use of information sources: the impact of quality and accessibility of information," *The Academy of Management Journal*, vol. 25, št. 4, str. 756–771, december 1982.

[41] B. Otto in ostali, "Towards a Framework for Corporate Data Quality Management," v zborniku *Proceedings of 18th Australasian Conference on Information Systems, 5–7 December 2007, Toowoomba, Australia.* Toowoomba: The University of Southern Queensland, 2007, str. 916–926.

[42] C. Parent in S. Spaccapietra, "Issues and Approaches of Database Integration," *Communications of the ACM*, vol. 41, št. 5, str. 166–178, maj 1998.

[43] T. R. Peltier, *Information Security, Policies, Procedures, and Standards: Guidelines for Effective Information Security Management*. Boca Raton: Auerbach Publications, CRC Press, 2002.

[44] G. Pernul, A. M. Tjoa in W. Winiwarter, "Modelling data secrecy and integrity," *Data & Knowledge Engineering,* vol. 26, št. 3, str. 291–308, Elsevier, julij 1998.

[45] E. Rahm in H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Engineering Bulletin,* vol. 23, št. 4, str. 3–13, december 2000.

[46] T. C. Redman, *Data Quality for the Information Age*. Artech House Inc, 1996.

[47] T. C. Redman, "The impact of poor data quality on the typical enterprise," *Communications of the ACM*, vol. 41, št. 2, str. 79–82, februar 1998.

[48] J. S. Richters in C. A. Dvorak, "A framework for defining the quality of communications services," *IEEE Communications Magazine*, vol. 26, št. 10, str. 17–23, oktober 1988.

[49] Z. G. Ruthberg, W. T. Polk, *Report of the Invitational Workshop on Data Integrity*, Washington: National Institute of Standards and Technology, 1989.

[50] F. Sadri, "Integrity Constraints in the Information Source Tracking Method," *IEEE Transactions on Knowledge and Data Engineering,* vol. 7, št. 1, str. 106–119, februar 1995.

[51] S. F. M. Sampaio, C. Dong in P. Sampaio, "DQ^2S – A framework for data quality-aware information management," *Expert Systems with Applications*, vol. 42, št. 21, str. 8304–8326, Elsevier, november 2015.

[52] M. Scannapieco in ostali, "The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems," *Information Systems – Special issue: Data quality in cooperative information systems*, vol. 29, št. 7, str. 551–582, Elsevier, oktober 2004.

[53] V. C. Storey, R. M. Dewan in M. Freimer, "Data quality: Setting organizational policies," *Decision Support Systems*, vol. 54, št. 1, str. 434–442, Elsevier, december 2012.

[54] V. C. Storey, C. B. Thompson in S. Ram, "Understanding database design expertise," *Data & Knowledge Engineering,* vol. 16, št. 2, str. 97–124, Elsevier, avgust 1995.

[55] M. Suer in R. Nolan. (12. januar 2015). Using COBIT 5 to Deliver Information and Data Governance. *COBIT Focus* [Online]. *S*tr. 1–6. Dostopno na: http://www.isaca.org/cobit/focus/pages/using-cobit-5-to-deliver-information-and-data-governance.aspx .

[56] K. Unsworth in ostali, "Goal hierarchy: Improving asset data quality by improving motivation," *Reliability Engineering and System Safety*, vol. 96, št. 11, str. 1474–1481, Elsevier, november 2011.

[57] F. Wan in ostali, "A Data Processing Middleware Based on SOA for the Internet of Things," *Journal of Sensors*, vol. 2015, ID članka 827045, 8 strani, Hindawi, 2015.

[58] R. Y. Wang, M. P. Reddy in A. Gupta, "An object-oriented implementation of quality data products", v zborniku *Proceedings of the Workshop on information technologies and systems (WITS-'93), december 1993, Orlando, Florida*. 1993, str. 48–56.

[59] R. Y. Wang, M. P. Reddy in H. B. Kon, "Toward quality data: An attribute-based approach," *Decision Support Systems*, vol. 13, št. 3–4, str. 349–372, Elsevier, marec 1995.

[60] R. Y. Wang, V. C. Storey in C. P. Firth, "A framework for analysis of data quality research," *IEEE Transactions Knowledge Data Engineering*, vol. 7, št. 4 str. 623–640, avgust 1995.

[61] R. Y. Wang in D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems*, vol. 12, št. 4, str. 5–33, marec 1996.

[62] S. Watts, G. Shankaranarayanan in A. Even, "Data quality assessment in context: A cognitive perspective," *Decision Support Systems*, vol. 48, št. 1, str. 202–211, Elsevier, december 2009.

[63] K. Weber, B. Otto in H. Oesterie, "One size does not fit all – a contingency approach to data governance," *ACM Journal of Data and Information Quality,* vol. 1, št. 1, str. 1–27, junij 2009.

[64] P. Weill in J. W. Ross, *IT governance*. Boston: Harvard Business School Press, 2004.

[65] P. H. Williams, C. R. Marguiles in D. W. Hilbert, "Data requirements and data. sources for biodiversity priority area selection," *Journal of Biosciences,* vol. 27, št. 4, str. 327–338, julij 2002.

[66] P. Woodall, A. Borek in A. K. Parlikad, "Data quality assessment: The Hybrid Approach," *Information & Management,* vol. 50, št. 7, str. 369–382, Elsevier, november 2013.

[67] M. Zviran in C. Glezer, "Towards generating a data integrity standard," *Data & Knowledge Engineering,* vol. 32, št. 3, str. 291–313, Elsevier, marec 2000.

Ostali viri

[68] *Advisera – 27001 Academy* [Online]. Dostopno na: http://advisera.com/27001academy/knowledgebase/iso-27001-vs-iso-27002/ .

[69] *Archi, The Free ArchiMate Modelling Tool* [Online]. Dostopno na: http://www.archimatetool.com/ .

[70] *Archimate* [Online]. Dostopno na: http://www.opengroup.org/subjectareas/enterprise/archimate .

[71] H. van den Berg in ostali. (17. 11. 2007). *Archimate Made Practical* (verzija 2.0) [Online]. ArchiMate Foundation. Dostopno na: http://www.archimate.nl/content/bestanden/archimate_made_practical_2008-04-28.pdf .

[72] *DAMA International – The Global Data Management Community* [Online]. Dostopno na: https://www.dama.org/ .

[73] *DataCleaner 4.5, The premier data quality solution* [Online]. Dostopno na: http://datacleaner.org/ .

[74] *DataMatch* [Online]. Dostopno na: http://dataladder.com/data-matching-software/ .

[75] *Eclipse Luna* [Online]. Dostopno na: https://eclipse.org/luna/ .

[76] *European Commission Directorate-General for Justice and Consumers* [Online]. Dostopno na: http://ec.europa.eu/justice/data-protection/ .

[77] *G2 Crowd – The world's leading business software review platform* [Online]. Dostopno na: https://www.g2crowd.com/categories/data-cleansing-quality/products .

[78] *IBM* [Online]. Dostopno na: www.ibm.com .

[79] *Islovar – terminološki slovar informatike* [Online]. Dostopno na: http://www.islovar.org .

[80] *ISO/IEC 27001 – Information security management* [Online]. Dostopno na: http://www.iso.org/iso/home/standards/management-standards/iso27001.htm .

[81] *ISO/IEC 27002:2013* [Online]. Dostopno na: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54533

[82] IT governance institute, Isaca. (2007). Cobit 4.1. *IT governance institute, Isaca* [Online]. Dostopno na: http://www.isaca.org/knowledge-center/cobit/pages/downloads.aspx .

[83] IT governance institute, Isaca. (2012). Comparing COBIT 4.1 and COBIT 5. *IT governance institute, Isaca* [Online]. Dostopno na: https://www.isaca.org/COBIT/Documents/Compare-with-4.1.pdf .

[84] *IT governance institute, Isaca, COBIT 5* [Online]. Dostopno na: https://cobitonline.isaca.org .

[85] *ITIL – Information Technology Infrastructure Library* [Online]. Dostopno na: https://www.axelos.com/best-practice-solutions/itil .

[86] *Java* [Online]. Dostopno na: https://java.com/en/ .

[87] M. Krisper, "Planiranje in Organizacija Informatike v Poslovnih Sistemih," *Gradivo s predavanj POIPS na podiplomskem študiju*. Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2009.

[88] *Oracle* [Online]. Dostopno na: www.oracle.com .

[89] *Oracle9i Database Concepts, Release 2 (9.2)* [Online]. Dostopno na: https://docs.oracle.com/html/A96524_01/index.htm#C .

[90] *Payment Card Industry Data Security Standard (PCI DSS)* [Online]. Dostopno na: https://www.pcisecuritystandards.org/ .

[91] U. Podobnikar, "Sistem za zaznavo napak v podatkovni bazi," *Seminarska naloga pri predmetu Elektronsko poslovanje na podiplomskem študiju.* Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, oktober 2015.

[92] *PricewaterhouseCoopers* [Online]. Dostopno na: http://www.pwc.com/ .

[93] *SAP* [Online]. Dostopno na: www.sap.com .

[94] *Stunnel* [Online]. Dostopno na: https://www.stunnel.org .

[95] Uredba o upravnem poslovanju [Online].
Dostopno na: http://www.pisrs.si/Pis.web/pregledPredpisa?id=URED3602#

[96] *Zakon o varstvu osebnih podatkov (uradno prečiščeno besedilo) (ZVOP-1-UPB1)* [Online]. Dostopno na: https://www.uradni-list.si/ .

[97] *ZPIZ – spletna stran Zavoda za pokojninsko in invalidsko zavarovanje* [Online].
Dostopno na: http://www.zpiz.si/ .